VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Nástroj pro hodnocení kvality a porovnávání webových projektů

Tool for quality assessment and comparison of web projects

2014 Jiří Baldik

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jiří Baldik**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Nástroj pro hodnocení kvality a porovnání webových projektů
Tool for Quality Assessment and Comparison of Web Projects

Zásady pro vypracování:

Cílem této práce je sestavit nástroj pro hodnocení kvality a podobnosti webových projektů. Nástroj bude stahovat obsah webových stránek, kontrolovat validnost kódu a hledat podobné stránky v rámci jednoho projektu nebo konkurenčních webů. Analyzujte pozici a stav konkurence, doporučte vhodné změny a úpravy vlastního webového projektu se zaměřením na SEO, SEM a copywriting.
1. Nastudujte problematiku hromadného paralelního stahování obsahu webových stránek. Implementujte nástroj, který bude extrahovat relevantní informace pro analýzu webových projektů.
2. Nastudujte problematiku validace zdrojového kódu webové stránky a aplikujte ji na stažené weby.
3. Nastudujte problematiku copywritingu, analyzujte a doporučte vhodné rozmístění klíčových slov na webové stránce.
4. Extrahujte vhodné komponenty webových projektů (n-gramy atd.) pro další porovnání.
5. Navrhněte univerzální algoritmy pro nalezení podobností vlastních webových projektů nebo podobnosti konkurenčních webů.
6. Výsledky podobnosti vhodně vizualizujte v podobě průnikových množin, tabulek a grafů.
7. Na základě získaných dat a dalších externích informačních zdrojů ověřte funkčnost daného řešení v praxi.
8. Výsledek zdokumentujte, porovnejte s existujícími řešeními a doporučte další možnosti využití a rozšíření.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení o autorství

Prohlašuji, že tuto diplomovou práci jsem vypracoval samostatně. V práci jsou uvedeny veškeré laterální prameny a publikace, ze kterých jsem čerpal.

Ve Studénce dne: 28. 4. 2014 Podpis:

Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu Ing. Radoslavovi Fasugovi, Ph.D., za velmi dobré vedení diplomové práce, bez níž by nevznikla.

Abstrakt

Obsahem této práce je seznámení s problematikou SEO a copywritingu. Následuje popis implementace nástroje, který hodnotí a porovnává stránky právě z pohledu SEO a copywritingu. První částí práce je detailní popis funkčnosti vyhledávačů, tedy stahování, parsování, ukládání dat vyhledávači a vyhledávání v takto získaných datech. Po vysvětlení principu funkčnosti internetových vyhledávačů práce volně přechází k optimalizaci stránek pro vyhledávače, tedy k SEO a copywritingu. V této části jsou uvedeny informace, jak by měla být napsána stránka, aby byla pro vyhledávače co možná nejvíce atraktivní. Další částí práce je samotná implementace aplikace. Je kladen důraz na to, aby byla aplikace paralelní, proto jsou v této části popisovány problémy a řešení problému se kterými jsem se během implementace analyzátoru a porovnávače setkal. Následuje popis výstupu z aplikace, tedy prezenční vrstvy, která je oddělená od vrstvy logické. Předposlední části práce je objektivní porovnání vytvořené aplikace s již existujícími aplikacemi. Nakonec se práce dostává k závěru, kde jsou zhodnoceny dosažené výsledky a navrženy případné možnosti rozšíření či modifikace aplikace.

Klíčová slova

SEO analyzátor, SEO copywriting, on-page optimalizace, princip funkčnosti internetových vyhledávačů, paralelní stahování webových stránek, parsování webových stránek, porovnávání webových stránek, generování n-gramů, porovnávání n-gramů, Java, PHP, MySQL

Abstract

The goal of this work is introduction of SEO and copywriting. Follows describe of implementation of application which evaluates pages by SEO and copywriting. The first part of the thesis contains detailed specification of how search engines works, thus downloading, parsing, saving obtained data and searching in these data. After explanation of principle of functionality of search engines, thesis freely passes to the optimization of pages for search engines, thus to the SEO and copywriting. In that part are contained information of how should be page written to be attractive for search engines as much as possible. Next part of the thesis is implementation of application itself. The emphasis is placed on it so the application has been parallel, therefor in that part of thesis are described problems and their solutions with which I've met during the implementation. Follows describe of output of application, thus describe of presentation layer which is separated from the logical layer. Last but one part of the thesis is objective comparison of existing applications with implemented application. In the end, thesis gets to the conclusion, where achieved results and suggested appropriate expansions and modifications of application are evaluated.

Key words

SEO analyzer, SEO copywriting, on-page optimization, principle of functionality of search engines, parallel downloading of web pages, parsing of web pages, comparing of web pages, n-grams generator, n-grams comparison, Java, PHP, MySQL

Seznam použitých symbolů a zkratek

SŘBD – systém řízení báze dat
MySQL – typ SŘBD
IS – informační systém
SEO – Search Engine Optimization (optimalizace pro vyhledávače)
PHP – skriptovací programovací jazyk, pracující na straně serveru, slouží k vytváření dynamických stránek
PPC – Pay Per Click (platba za klik)
HTML – Hyper Text Markup Language (hypertextový značkovací jazyk), jazyk pro vytváření základní obsahové kostry webové stránky
On-page – faktory webové stránky ovlivňující umístění stránky ve výsledcích vyhledávání, tyto faktory lze ovlivnit editací zdrojového kódu stránky
Off-page – faktory webové stránky ovlivňující umístění stránky ve výsledcích vyhledávání, obsahují například odkazy směřující na danou stránku z jiných stránek
AJAX – změna obsahu webových stránek bez nutnosti opětovného načítání pomocí technologií HTML, JavaScript, DOM a XMLHttpRequest
DOM – objektově orientovaná reprezentace XML nebo HTML dokumentu
JavaScript – objektově orientovaný skriptovací jazyk pracující ve webovém prohlížeči na straně uživatele
XMLHttpRequest – Asynchronní výměna dat s webovým serverem
Java – objektově orientovaný programovací jazyk nezávislý na architektuře
URL – Uniform Resource Locator (jednotný lokátor zdrojů), slouží k přesné specifikaci umístění zdrojů a informací
C++ – multiparadigmatický programovací jazyk
ASP.NET – součást .NET Frameworku pro tvorbu webových aplikací
.NET Framework – zastřešující název pro soubor technologií v softwarových produktech
Python – dynamický objektově orientovaný skriptovací programovací jazyk

Seznam obsažených obrázků a tabulek

Obsah

1. Úvod

Cílem této diplomové práce je vytvořit program, který bude hodnotit SEO kvalitu webových projektů a následně tyto projekty mezi sebou porovnávat. Aplikace nebude nabízet jen porovnávání projektů mezi sebou, ale také porovnávání webového projektu vůči sobě samému, čímž může být odhalena například duplicita stránek. Konkrétně jde o hodnocení webových projektů psaných v programovacím jazyce HTML s obsahem v českém jazyce. I když někdo může namítat, že existují již podobné nástroje, jako je například WebSEO, tak nedostatkem těchto nástrojů v českém prostředí je právě absence podpory češtiny.

Od aplikace je požadováno, aby se skládala ze dvou na sobě nezávislých částí. První část aplikace poběží na pozadí Linuxového serveru a bude čekat na požadavek o stažení a zhodnocení webového projektu. Druhá aplikace slouží k zobrazení kvalit projektů a výsledků porovnávání jednotlivých projektů. Abychom tohoto docílili, budeme potřebovat mimo jiné systém řízení báze dat, který bude uchovávat a sdílet informace mezi těmito dvěma na sobě nezávislými aplikacemi.

Co se týče samotného budoucího uživatele této aplikace, je kladen důraz na jednoduchost nastavení analýzy. Z toho důvodu aplikace požaduje pouze informace o URL webového projektu ke zpracování, hloubky zanoření v daném projektu a významnost jednotlivých HTML elementů a atributů. Aplikace poté vytváří množství kombinací a testů, ze kterých vyvodí závěry. Uživatel si potom ve výsledku může vybrat mezi těmito kombinacemi, bez nutnosti opětovného stahování a zpracování celého projektu.

Dále je od aplikace požadováno, aby respektovala omezení pro roboty, konkrétně pravidla určené pro všechny roboty. Pokud platí omezení například pouze pro Google, aplikace jej sice zaregistruje a následně uživateli zobrazí, ale sama stránku stáhne a projde všechny její vnitřní odkazy. Jde tedy o to, nevytvářet špionážní software, nýbrž software, který se snaží nabídnout uživateli pohled na jeho webový projekt, tak jak ho pravděpodobně vidí vyhledávače. Pravděpodobně pro to, protože vyhledávače zveřejňují jen minimum informací o tom, jak ve skutečnosti fungují.

K samotné funkčnosti aplikace. Aplikace stáhne zadanou URL adresu, získá z ní všechny vnitřní odkazy, obsahy z elementů, které jsou z hlediska SEO významné (jedná se tedy o nadpisy, odstavce, zvýrazněný text, apod.), zkontroluje, zda je HTML kód stránky validní, zda text na stránce neobsahuje chyby, apod. Získané informace následně uloží do databáze. V paměti si však ponechá text stránky, jelikož s tímto textem bude později ještě dále pracovat, při generování klíčových slov. Tohle cyklicky provádí pro všechny nalezené vnitřní odkazy domény, dokud buďto neprojdeme všechny stránky domény nebo dokud nedosáhneme omezení hloubky procházení.

Nyní tedy ke generování klíčových slov stránky neboli n-gramů velikosti jedna, dva a tři. Tyhle velikosti jsou zvoleny záměrně, jelikož Google prohlašuje, že on sám nevytváří větší n-gramy. Před vytvořením n-gramů jsou z textu odstraněny nežádoucí znaky a stop slova, která by mohly znehodnotit výsledný n-gram. N-gramy jsou následně uloženy do databáze, do slovníků a identifikační čísla ze slovníků jsou uložena do databáze stránek, kde zároveň obsahují ve kterém elementu nebo atributu se

nacházejí a s jakou četností. V databázi ovšem není jen jediný slovník, nýbrž hned 12. Důvodem je, že nejsou vytvářeny pouze n-gramy, které se skutečně nacházejí na webu, ale také n-gramy dle následujících kombinací pravidel:

- Slova v n-gramech jsou/nejsou seřazena podle abecedy
- Slova v n-gramech se skládají/neskládají pouze z kořenů slov
- Slova ve slovnících respektují/ignorují synonyma

N-gramy se samozřejmě vytvářejí pro každou stránku zvlášť a následně jsou stránky, respektive n-gramy stránek mezi sebou porovnávány. Vzhledem k tomu, že aplikace zná četnost jednotlivých n-gramů a ve kterém elementu nebo atributu se na stránce nacházejí, tak díky výše zmíněnému po uživateli požadovanému definování významnosti jednotlivých elementů a atributů, je vypočítána podobnost jednotlivých stránek. Zároveň aplikace určí, který n-gram na stránce dominuje, neboli na které slovo je stránka optimalizována z pohledu vyhledávače.

V aplikaci zobrazující výsledek si tedy uživatel může projít SEO kvality jednotlivých stránek, které jsou pro zpřehlednění ohodnoceny vlastním bodovacím systémem. Tento bodovací systém zohledňuje všeobecně známá i ty méně známá SEO pravidla či duplicitní obsah. Dále si může uživatel projít výskyt a četnost jednotlivých klíčových slov na stránce, tedy zdali je stránka optimalizovaná pro klíčové slovo, jaké uživatel zamýšlel. Nebo může uživatel porovnávat klíčové slova a jejich četnost mezi jednotlivými stránkami, či stránkami různých projektů.

2. SEO

2.1 Co je to SEO?

2.1.1 Představení

Zkratka SEO je odvozena z anglického názvu „Search Engine Optimization“, tedy „optimalizace pro vyhledávače“. Už ze samotného názvu, je více či méně zřejmé, o co se vlastně jedná.

Pokaždé když je do jakéhokoliv vyhledávače, ať už Google, Yahoo!, bing či seznam, zadána vyhledávací fráze a vyhledávač vrátí seznam stránek odpovídající zadanému výrazu. Uživatelé mají normálně tendenci navštěvovat stránky, které se jim objeví na začátku tohoto seznamu, jelikož předpokládají, že tyto stránky budou pro jejich dotaz relevantnější. To, co se skrývá za umístěním stránek takto vysoko ve výsledku vyhledávání je právě optimalizace pro vyhledávače, neboli SEO, jenž se postupně stalo samostatným marketingovým odvětvím.

SEO tedy může být definováno jako technika, která pomáhá vyhledávačům najít a ohodnotit optimalizovanou stránku lépe než milióny ostatních stránek v odpovědi na specifický vyhledávací dotaz. SEO tedy pomáhá zvětšit přísun uživatelů z výsledku vyhledávání u konkrétního vyhledávače na optimalizovanou stránku.

2.1.2 Jak vyhledávače fungují?

Ač to může znít směšně, tak první věc, kterou si u SEO musíme uvědomit je fakt, že vyhledávače jsou stroje, ne lidé. Důvodem, tohoto zamyšlení je, že stroje a lidé se na webové stránky dívají odlišně. Naproti lidem, vyhledávače zajímá pouze text. Ačkoliv technologie jdou rychle kupředu, tak vyhledávače jsou stále daleko za inteligentními bytostmi, které mohou například ocenit krásu nádherného designu stránky, či si vychutnat zvuky a pohyby ve videích na stránkách. Místo toho, vyhledávače crawlují web a hledají na něm konkrétní prvky stránky (hlavně text), které jim prozradí o čem je obsah stránky. Tato krátká specifikace není nejpřesnější, ale podrobnější popsání jednotlivých kroků vyhledávačů pro poskytnutí relevantních výsledků se objeví v průběhu této práce – *crawlování*, *indexace*, *zpracování*, *výpočet relevance* a *vracení výsledků*.

Kapitola 2.2 – Architektura vyhledávače

Jelikož v současnosti je největším světovým vyhledávačem Google, a u nás se již taky dostal na trůn, kde vystřídal český přední vyhledávač Seznam, tak se v tato práce bude soustředit zejména na popis technologií a technik Googlu. Týká se to jak architektury systému, tak funkcí dílčích programů systému.

Kapitola 2.3 – Web crawler

Jako první ze všeho vyhledávače crawlují/stahují web aby zjistili, co se na dané stránce nachází. Tento úkol je zajištěn programem, který se nazývá *crawler* nebo také *spider* (či v konkrétních případech

konkrétních vyhledávačů například Googlebot pro Google, či SeznamBot pro Seznam). Crawlery sledují odkazy z jednotlivých stránek na druhé a indexují vše, na co narazí. Rozsáhlost internetu přes 20 miliard stránek nedovoluje, aby crawler navštívil stránku několikrát denně, proto aby se ujistil, zda se obsah neaktualizoval. Někdy se může stát, že daný crawler nenavštíví stránku i měsíc, či dva.

Kapitola 2.4 – Indexer

Poté co je stránka stažená, je dalším krokem indexace jejího obsahu. Zaindexovaná stránka je uložena v obrovské databázi odkud může být později opětovně získána. Indexace je v podstatě proces identifikace slov a výrazů, které nejlépe vystihují obsah stránky a přiřazení této stránky ke konkrétním výrazům. Pro člověka by bylo v podstatě nemožné vypořádat se s tak obrovským množstvím informací, ale vyhledávače tohle hravě zvládají. Občas se však stane, že nevystihnou obsah stránky správně. Tvůrci webových stránek můžou vyhledávačům pomoci se správným vystižením obsahu pomocí SEO. Tím si zajistit vyšší ohodnocení u daného vyhledávače.

Kapitola 2.5 – Query procesor

Když je zadán vyhledávací dotaz, vyhledávač jej zpracuje, neboli porovná zadaný řetězec ve vyhledávacím dotazu se zaindexovanými stránkami v databázi. Jelikož je pravděpodobné, že více než jedna stránka (většinou milióny stránek) obsahuje požadovaný řetězec, vyhledávač musí přistoupit k dalšímu kroku, jímž je výpočet relevance jednotlivých stránek vracených z indexu na zadaný dotaz.

Existuje mnoho algoritmů na výpočet relevance, které si vyhledávače bedlivě střeží, jelikož prozrazením těchto algoritmů by vývojářům webových stránek poskytli návod jak se dostat na první příčku ve výsledcích vyhledávání. Každý z těchto algoritmů obsahuje vlastní ohodnocení významnosti jednotlivých prvků stránky, jako klíčové slova, meta tagy, či odkazy na stránce. Tohle je důvod, proč jednotlivý vyhledávače vracejí různé výsledky pro stejný dotaz. Navíc, je známý fakt, že přední světové vyhledávače jako Google, Yahoo!, či Bing pravidelně obměňují jejich algoritmy. Pokud se tedy chce stránka udržet v předních příčkách, musí se na tyto změny adaptovat.

Posledním krokem vyhledávačů je vracení výsledků. Nejedná o nic víc než zobrazení výsledků v prohlížeči, čili zobrazení konečného množství stránek pro zadaný dotaz seřazených od nejrelevantnějších po nejméně relevantní.

2.1.3 Rozdíly mezi předními vyhledávači

Ačkoliv základní princip vyhledávání a crawlování je u všech vyhledávačů stejný, tak malé rozdíly mezi nimi vedou k velkým rozdílům v relevantnosti výsledků. Pro různé vyhledávače jsou důležité odlišné faktory. Kdysi si lidé pohybující se v prostředí SEO dobírali algoritmus od Bingu že je záměrně vytvořený tak, aby vracel přesně opačné výsledky než Google. Tohle je samozřejmě nesmysl, ale napovídá to o tom, k jak velkým rozdílům v hodnocení relevantnosti může ve výsledku docházet. Z toho se dá vyvodit jeden podstatný závěr a to, že pokud jsou stránky optimalizovány, musí být určeno, pro který vyhledávač se provádí optimalizace. Pokud pokut je stránka optimalizována pro všechny vyhledávače, je práce mnohem náročnější. Navíc nemůže být teoreticky optimalizace nikdy úspěšná a stane se to, že sice budou o stránce vědět všechny vyhledávače, pravděpodobně ji budou

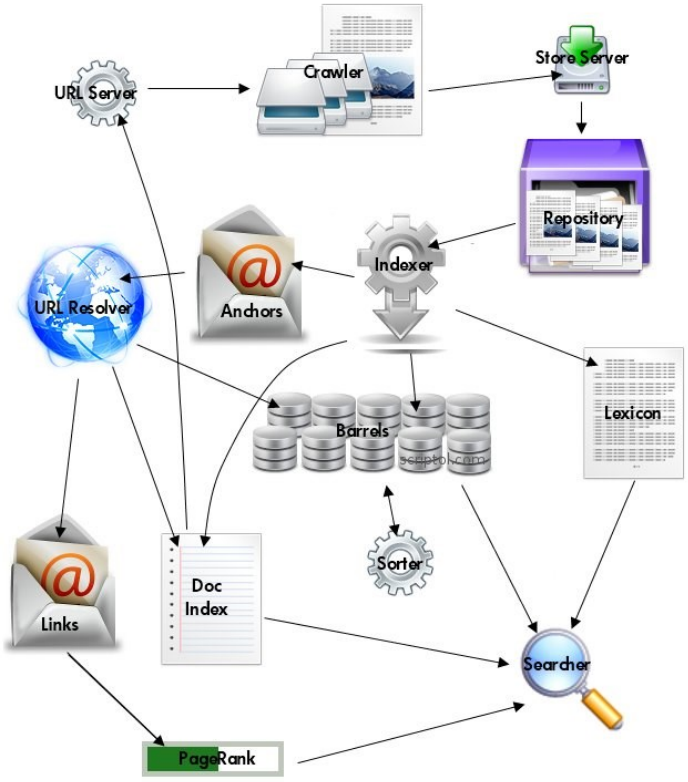
hodnotit i kladně, ale u žádného vyhledávače nebude stránka dostatečně vysoko na to, aby to přineslo tížený výsledek.

Existuje mnoho příkladů pro objasnění rozdílů mezi jednotlivými vyhledávači. Například, pro Yahoo! a Bing, jsou on-page faktory klíčovými prvky, zatímco pro Google jsou velice důležité odkazy. Dále pro Google jsou starší stránky relevantnější než nové, zatímco Yahoo! neupřednostňuje starší stránky před novějšíma. Stránka tedy bude pravděpodobně potřebovat více času, aby se dostala na první příčky v Googlu, v porovnání s Yahoo!.

2.2 Architektura vyhledávačů

2.2.1 Přehled

Jak bylo již řečeno, práce je zaměřená především na popis principu fungování Googlu. Je to dáno také tím, že většina vyhledávačů čerpá nápady právě od Googlu.



Obrázek 1: Architektura vyhledávače

V této sekci bude stručně představena architektura systému, která je nastíněna na obrázku 1 [10]. V dalších sekcích budou podrobně rozebírány aplikace a datové struktury systému. Za zmínku také stojí, že většina aplikací Googlu je kvůli efektivnosti implementovaná v C nebo C++ a můžou běžet buďto v systému Solaris nebo v Linuxu.

Google provádí crawlování (stahování stránek) je prováděno několika crawlery. Je zde URL server, který posílá crawlerům seznamy URL adres, které mají být staženy. Stránky, které jsou staženy crawlery, jsou poté poslány do tzv. storeserveru. Storeserver potom zkomprimuje a uloží stránku do repositáře. Každá stránka má přiřazené jedinečné identifikační číslo, tzv. docID, které je přiřazeno pokaždé při získání nové URL adresy ze stránky. Indexování je zajištěno tzv. indexerem a sorterem. Indexer provádí mnoho funkcí, například přečte repositář, vybere si dokument, dekomprimuje jej a rozparsuje, tedy „vytáhne“ si z dokumentu jen to, co je pro něj podstatné. Každý dokument je převeden do sad výskytů slov, tzv. hitů. Hit obsahuje samotné slovo, jeho pozici v dokumentu, které slova se kolem něj vyskytují, odhad velikosti, zvýraznění písma, apod. Indexer dále hity rozděluje do „barelů“, čímž vytváří částečně seřazený „forward index“ (forward index proto, že existuje ještě jeden index, který se nazývá „inverzní index). Indexer provádí další důležitou věc, jíž je vyparsování (nalezení, vytažení) odkazů z každé stránky a ukládá významné informace o každém tomto odkazu do anchor souborů. Tyto anchor soubory obsahují informace odkud a kam daný odkaz směřuje a text, kterým je odkazován, tzv. anchor text.

Poté přichází na řadu URLresolver, který čte anchor soubory, převádí relativní odkazy na absolutní a přiřadí k odkazu docID. Ať už nové, či docID stránky, která již existuje v databázi. Vkládá anchor text do forward indexu, přidružený k docID na který anchor text odkazuje. Také generuje databázi odkazů, což jsou páry docID (odkud – kam). Databáze odkazů je poté využívána pro výpočet PageRanků pro všechny stránky.

Tzv. sorter si přebírá barely, které jsou seřazené dle docID a přetřídí je podle wordID pro následné vygenerování inverzního indexu. Sorter dále vytváří seznam wordID a offsetů do inverzního indexu. Program zvaný DumpLexicon dá tyto seznamy vytvořené indexerem dohromady a vygeneruje nový lexikon, který bude používaný při vyhledávání. Vyhledávač, který běží na webovém serveru, používá tyto lexikony, vytvořené programem DumpLexicon spolu s inverzním indexem a PageRankem k dopovězení na vyhledávací dotaz. [9]

2.2.2 Hlavní datové struktury

Datové struktury Googlu jsou optimalizovány pro práci s velkou kolekcí dokumentů, aby mohly být tyto data rychle procházeny, indexovány a vyhledávány. Ačkoliv vývoj procesorů jde neustále kupředu a za poslední roky obrovsky pokročil, disky jsou na tom podstatně hůře. I když vidina lepší budoucnosti by mohla být v SSD discích. SSD disky však momentálně nejsou pro servery nejvhodnějším kandidátem, vzhledem k jejich relativně vysoké cenně v porovnání s klasickými disky. Google je navržený k tomu aby umožnil vyhledávání kdykoliv je to jen možné což má značný vliv na návrh datové struktury systému. [9]

Repositář

Repositář obsahuje celý HTML kód každé stránky. Každá stránka je komprimovaná. Dokumenty jsou uloženy jeden po druhém a jsou identifikovány jedinečným číselným označením, tzv. docID. Dále je k těmto docID přiřazena délka dokumentu a URL adresa dokumentu.

Index dokumentu

Index uchovává informace o každém dokumentu. Informace uložené v každém záznamu obsahují současný stav dokumentu, ukazatel na dokument v repositáři, checksum dokumentu, apod. Záznam obsahuje také ukazatel na proměnlivý dokument zvaný docinfo, který obsahuje URL a title dokumentu (pokud byl dokument již crawlován). Pokud ještě nebyl dokument crawlován, obsahuje index ukazatel na URLlist, který obsahuje pouze URL.

Dále je v indexu soubor, který je používán pro převod mezi URL a docID. Je to seznam URL checksumů a k nim jejich příslušné docID, který je seřazen právě podle checksumů. Při vyhledávání docID příslušící dané URL adrese, je URL adresa převedená na checksum a metodou binárního vyhledávání nalezena v seznamu.

Lexikon

Program zvaný DumpLexicon přebírá od sorteru seznam seřazený podle wordID, plus další lexikon vytvořený indexerem. Spojením těchto dvou lexikonů vytvoří lexikon pro vyhledávání.

Hit listy

Hit list odpovídá seznamu výskytu daného slova v daném dokumentu, obsahující pozici, font, a velikost písmen. Hit list je zodpovědný za zabrání většiny paměti jak ve forward indexu tak inverzním indexu.

Forward index

Forward index je po částech uložen v barelech. Každý barel obsahuje seznam wordID. Pokud dokument obsahuje slova spadající do konkrétního barelu, je do tohoto barelu vloženo docID daného dokumentu, následováno seznamem wordID s hit listem příslušícím těmto slovům. Tento index zabírá relativně dost místa vzhledem k tomu, že se jednotlivé docID vyskytují duplicitně ve více barelech.

Slovo	Dokument
pes	Dokument 1, Dokument 2, Dokument 3, Dokument 4, Dokument 5
dělá	Dokument 1, Dokument 3, Dokument 5
haf	Dokument 1

Tabulka 1: Forward index

Inverzní index

Inverzní index se skládá ze stejných barelů jako forward index, mimo to, že tyto barely byly již zpracovány sorterem. Lexikon obsahuje ukazatel na barel, do kterého spadá každé wordID. Ukazuje na doclist daného wordID spolu s jejich příslušnými hit listy. Tento doclist reprezentuje všechny výskyty daného slova ve všech dokumentech.

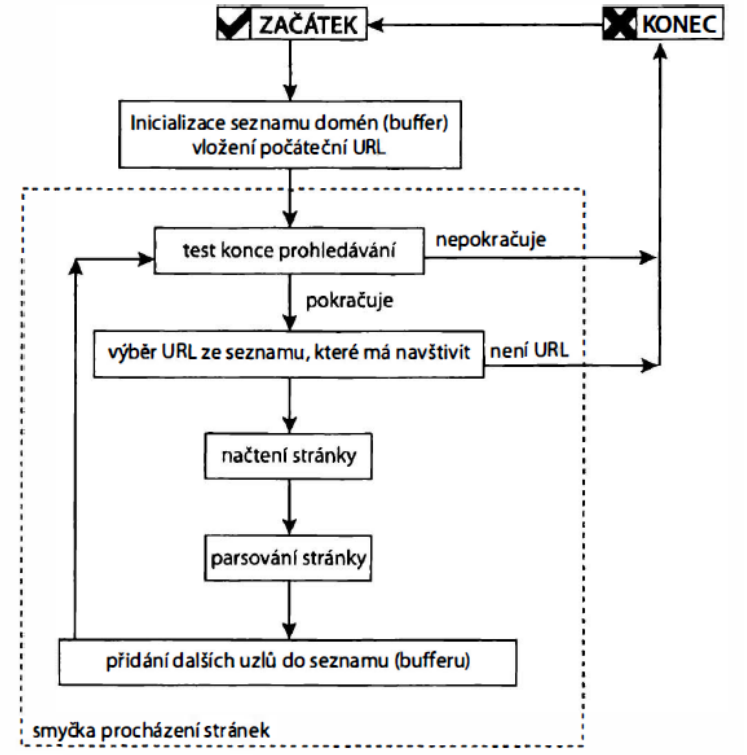
Slovo	Dokument
Dokument 1	pes, dělá, haf
Dokument 2	pes
Dokument 3	pes, dělá
Dokument 4	pes
Dokument 5	pes, dělá

Tabulka 2: Inverzní index

2.3 Web crawler

2.3.1 Představení

Nejprve stručné objasnění, co to vlastně web crawler je. Crawler není oficiální výraz pro tento program, k nalezení jsou stejné definice pod klíčovými slovy spider, robot, bot, či české výrazy jako například pavouk, apod. Práce se však bude držet výrazu crawler.



Obrázek 2: Základní princip crawleru

Stručně se tedy jedná o program, který stahuje webové stránky, nejčastěji pro internetové vyhledávače. Crawler začne svou práci počáteční seznamem URL adres ke stažení. Nejprve vloží URL adresy do fronty, kde je seřadí dle své vlastní prioritizace. S této fronty poté crawler postupně vybírá URL adresy ke stažení. Stáhne obsah stránky nacházející se na dané URL adrese, vyextrahuje všechny URL adresy z této stránky a zařadí je do fronty. Tento proces se opakuje tak dlouho, dokud se crawler

neukončí, ať už z důvodu vyčerpání všech URL adres z fronty, či dosáhnutí limitu možného stažení stránek, apod. Každá stažená stránka je dále předávána dalším programům pro zpracování. Například programům pro analýzu obsahu, indexace stránky, apod. [1]

Běžný uživatel se s crawlery setkává v prostředí internetu doslova denně a jeho efektivní práce s internetem je na těchto programech nesmírně závislá. Jak již bylo zmíněno, nejčastějším užitím crawlerů jsou právě internetové vyhledávače. Nebýt těchto programů, jen velmi těžko by byly vyhledávány informace v prostředí internetu. Dá se tedy říct, že uživatel internetu se nepřímo setkává s crawlerem pokaždé, když použije jakýkoliv internetový vyhledávač. Proč nepřímo? Protože internet obsahuje obrovské množství informací, které není žádný současný program prohledávat v reálném čase, proto v pozadí internetových vyhledávačů běží crawleři, kteří prohledávají internet a ve spolupráci s ostatními programy pro analýzu obsahu webových stránek udržují svou vlastní databázi s informacemi (neboli index) aktuální. Je tedy zřejmé, že při používání vyhledávače není prohledáván přímo internet, ale pouze databáze obsahující více či méně relevantní informace k jednotlivým dokumentům na internetu.

Další možností kdy se může uživatel setkat s crawlerem je, pokud je vlastníkem jakékoliv webové stránky. Je velice pravděpodobné, že jeho stránku nějaký ten crawler navštívil, nebo navštěvuje. Ovšem je důležité, aby se tento crawler o této stránce nějakým způsobem dověděl, ať už na URL adresu narazí na nějaké jiné webové stránce, či dostane přímou informaci kde onu stránku nalézt.

Dále existují tzv. osobní crawleři, jako je například program WebAttache, kdy se jedná o off-line prohlížeč, do kterého jsou zadány URL adresy zájmu, crawler je stáhne a program následně zobrazí jejich obsah. Aplikace zvládá i vlastnosti jako je například vyhledávání.

Vzhledem k tomu, že obsah internetu je v dnešní době obrovský a neustále roste, návrh opravdu kvalitního crawlera musí čelit mnoha výzvám. Dle určitých studií současný internet obsahuje přes 20 miliard stránek, jejichž velikost je okolo 320KB, přičemž Google má zanalyzovaných okolo 4,7 miliard z těchto stránek [8]. Mimoto, nejenže počet stránek neustále narůstá, ale navíc již existující stránky jsou neustále aktualizovány. To v podstatě znamená, že crawler musí neustále procházet celý internet, aby jak zajistil zařazení nových stránek do indexu, tak aktualizaci již stažených stránek v indexu. Proto se crawlery řídí různými pravidly, například pravidly významnosti jednotlivých stránek, jak často jsou stránky aktualizovány respektive jak často je zapotřebí je dané stránky navštěvovat, apod.

2.3.2 Výzvy implementace crawlera

Které stránky má crawler stahovat?

Ve většině případů nemůže crawler procházet všechny stránky na internetu. Dokonce i ty nejrozsáhlejší internetové vyhledávače mají v současné době za indexovaný jen malý zlomek celého internetu. Proto je velice důležité pro crawlery pečlivě vybírat stránky a procházet „důležité“ přednostně.

Jak má crawler aktualizovat stránky?

Po stažení určitého počtu stránek crawlerem musí crawler zajistit aby tyto stránky byly aktuální. Musí tedy tyto stránky opětovně navštěvovat. Vzhledem k tomu, že každá stránka se mění v jiném časovém intervalu, crawler musí pečlivě zvážit kdy a jakou stránku znovu navštívit aby zajistil nejvyšší aktuálnost relevantnost své databáze. Například, pokud se nějaká stránka aktualizuje jen zřídka kdy, musí crawler zajistit aby ji nenavštěvoval tak často jako ostatní stránky. [7]

Jak by měl crawler minimalizovat zatížení webů?

Když crawler stahuje stránky z webu, zatěžuje také jiný server nežli jen svůj vlastní. Jinými slovy, zpomaluje chod webu pro běžné uživatele. Například, pokud crawler stahuje stránku S na webu W, server musí poslat stránku S crawleru, což zatěžuje jeho disk a procesor. Po nalezení musí být stránka odeslána přes síť, což je další zatěžovaný zdroj sdílený vícero organizacemi. Proto by měl správný crawler minimalizovat dopad na tyto zdroje. Pokud tak neučiní, může se stát, že si bude provozovatel webu, či serveru stěžovat či dokonce zakáže přístup danému crawleru na jeho server/web. [7]

Jak paralyzovat crawlerování?

Vzhledem k rozsáhlosti internetu, crawler obvykle běží na vícero zařízeních a stahuje stránky paralelně. Tato paralelizace je nezbytná pro stažení velkého počtu stránek v přijatelném čase. Samozřejmě jednotliví crawleři musí být koordinování, aby neprocházeli stránky současně nebo více krát než je požadováno. Na druhou stranu tato koordinace přináší další problémy, jako komunikaci mezi jednotlivými crawlery a nebo omezení počtu crawlerů.

2.3.3 Jak usnadnit crawlerovi práci

[2] Je zřejmé, že aby si vyhledávač zařadil novou stránku do databáze, musí se o ní nějakým způsobem dozvědět. Existuji dva způsoby. První byl již zmíněn, jde o způsob, dalo by se říci přirozený, kdy crawler narazí na novou URL adresu někde na internetu, respektive na stránce kterou zrovna prochází. Druhým způsobem je, když je vyhledávači předána informace o nové URL adrese, kterou má projít. Každý vyhledávač má k tomuto účelu specializované stránky, v případě Googlu se jedná o stránku *https://www.google.com/webmasters/tools/submit-url*. Zadání nové URL adresy prostřednictvím těchto stránek však nijak nezaručuje jejich umístění ve výsledcích vyhledávání. Navíc se většina odborníků shoduje na tom, že první možnost, tedy kdy crawler narazí na URL adresu někde na internetu, je efektivnější, jelikož crawlerovi nezbývá nic jiného, než následovat nově nalezený odkaz.

Co crawler na stránce uvítá

Kvalitní obsah

- Na stránkách by se měly vyskytovat slova, které může vyhledavač zahrnout do výsledků svého vyhledávání. Což je logické, jelikož pokud uživatel chce docílit toho, aby se stránka zobrazovala ve výsledcích vyhledávání na konkrétní výraz, musí zajistit, aby se tento výraz na stránce opravdu nacházel. Nemůže se tedy spoléhat na to, že má výraz obsažený ve videu, či obrázku. Další části této práce se budou podrobněji zabývat tím, kde by se měl, či neměl výraz, na který je stránka optimalizována nacházet.

- Velikost stránky by neměla přesahovat více než 100KB, respektive velikost HTML kódu, bez obrázků. Důvodem je především to, že pokud je stránka větší, předpokládají vyhledávače, že načtení takovéto stránky bude trvat delší dobu, a doba načítání stránky je z hlediska SEO velice důležitá. Neměla by být větší než 3 sekundy. Nicméně doby, kdy Google neukládal do své databáze velké stránky, jsou již minulostí.
- Validní HTML kód. I když vyhledávače na tomhle příliš nelpí, takže se v prostředí internetu vyskytují stránky, které se nacházejí na prvním místě ve výsledku vyhledávání a nejsou validní, tak má validace pořád smysl. Může totiž nastat případ, kdy vyhledávač přečte kód jinak, než bylo zamýšleno, právě v důsledku nevalidního HTML kódu

„Pěkné“ URL adresy

- Jedná se o tzv. SEO friendly URL adresy. Což znamená, že zaprvé adresa nebude obsahovat příliš mnoho parametrů (jelikož crawlery většinou neprocházejí stránky, které mají v URL adrese více než 4 parametry) a zadruhé, parametry nejsou zapsány ve formátu *?parametr1=hodnota1&parametr2=hodnota2*, nýbrž ve formátu */hodnota1/hodnota2*, čehož se může docílit například modifikací .htaccess souboru. Výsledná adresa by poté mohla vypadat například následovně: *www.domena.cz/stranka1/podstranka1*

Kvalitní a silnou navigaci

- Všechny odkazy na stránce, jak interní tak externí, by měly být funkční. Každý nefunkční odkaz stránku z hlediska SEO ohodnocení znehodnocuje.
- Všechny stránky, které chceme mít na webu dostupné, musí být dosažitelné alespoň jedním statickým odkazem. Pokud tomu tak není, a na webu existují odkazy na stránky, které jsou generovány „za běhu“ (buďto pomocí JavaScriptu nebo AJAXu), pravděpodobně nastane situace, kdy vyhledávač stránky za těmito odkazy nezařadí do výsledků vyhledávání. Důvod je prostý a to, že generované odkazy jsou pro crawler v drtivé většině případů neviditelné.
- Je vhodné, pokud web obsahuje tzv. mapu web. Jedná se o stránku, na které má uvedeny odkazy na všechny podstránky na webu. Zamezí se tak tomu, že by se vyhledávač na některou stránku nedostal. Navíc se tím také docílí toho, že se vyhledávač dostane na každou stránku maximálně přes dva odkazy. Proč je tohle důležité? Protože vyhledávače nepřidají na hodnocení stránkám, které jsou dosažitelné přes tři a více odkazů, neboli stránky, které jsou zanořeny hluboko na webu.
- Doména by měla pokaždé obsahovat soubor sitemap.xml, který obsahuje seznam stránek a k ním údaje, jako například jak často je stránka modifikována, či kdy byla naposledy modifikována. Tímto se ulehčí práce crawlerům, jelikož si přes soubor sitemap.xml můžou jednoduše zjistit, že byla stránka modifikována a aktualizovat svou databázi. Neboli navštívit stránku znovu. Zároveň tím, že uvedeme předpokládanou frekvenci aktualizování, můžeme zajistit pravidelné návštěvy crawlerů, které se budou snažit udržovat databázi vyhledávače aktuální.

Když nebloudí

- Pokud je stránka přesunuta na jinou URL adresu, měli by původní stránka přesměrovávat na novou. Zamezí se tak situaci, kdy by crawler zamířil na neexistující stránku.
- Při přesunu je taktéž vhodné crawlery upozorňovat na fakt, zda se jedná pouze o dočasné přesunutí stránky či permanentní.
- Definování stránek, na které robot nesmí vstupovat je taktéž nezbytnou součástí kvalitního webu. K tomuto účelu slouží buďto soubor robots.txt, umístěný v kořenovém adresáři webu, informace přímo u odkazů na stránce (atributu `rel` s hodnotu `nofollow`) a `meta` element `robots` v hlavičce stránky. Jak konkrétně můžou být definovány omezení pro roboty je detailně popsáno v dalších částech práce.

Zakázaní přístupu crawlerovi na konkrétní stránky

Existují tři způsoby, jak zamezit crawlerům přístup na danou stránku. Zmíněny budou všechny tři způsoby, nicméně poslední způsob je ten, na který se práce zaměří a které jsou v praxi nejvyužívanější.

- První možností je, že se nikde na internetu nebude vyskytovat odkaz vedoucí na stránku, která má být pro crawlery zakázána. Nicméně, tohle není zrovna nejšťastnější způsob, jelikož nemlže být nikdy stoprocentně zaručeno, že nikdo někde v diskuzním fóru nebo na svých stránkách neuvede odkaz této stránky.
- Další možností, opět ne příliš nejšťastnější zato spolehlivější je, pokud bude URL adresa stránky obsahovat velký počet parametrů. Jak již bylo zmíněno, crawleři se vyhýbají stránkám s více než čtyřmi parametry v URL adrese.
- Nejspolehlivější metodou je přímo zakázat robotům přístup na konkrétní stránku. Toho lze docílit buďto pomocí souboru robots.txt, `meta` elementu `robots` nebo pomocí atributu `rel` přímo v odkazu na stránce, tedy elementu `a`.

Proč vůbec zakazovat robotům přístup na stránky? Důvodů je hned několik. Může se jednat o:

- Placené články ve zpravodajských archivech
- Interní diskuzní fóra
- Výsledky vyhledávání položek v e-shopu, či na běžném webu
- Stránky pro tisk (zabrání se tak duplicitnímu obsahu)
- Apod.

Meta robots

Jedná se o hlavičkový meta element, který vypadá následovně `<meta name="robots" content="..."/>`, kde může atribut `content` nabývat následujících hodnot:

- `index` nebo `noindex` – robot smí, či nesmí indexovat danou stránky, tedy zařadit ji do své databáze.
- `follow` nebo `nofollow` – robot smí, či nesmí následovat odkazy uvedené na dané stránce, ať už vnitřní nebo vnější.

- `all` – má stejný význam jako „`index, nofollow`“. Prázdná hodnota má stejný význam jako `all`.
- `none` – stejný význam jako „`noindex, nofollow`“, což pro robota znamená, že má stránku absolutně ignorovat

Co se týče efektivity této metody, není pro robota zrovna nejlepší. Robot totiž musí stránku navštívit, rozparsovat a až poté zjistí, že nemá stránku indexovat. Ba co víc, robot se musí na stránku neustále vracet, aby zjistil, zda se náhodou instrukce nezměnila. Je tedy jasné, že moc práce crawlerovi tímto řešením zrovna neušetříme a zatížení serveru se taky nezmenší.

Rel nofollow v odkazu

Pomocí atributu `rel` s hodnoto `nofollow` můžou být na stránce označeny ty odkazy, které nemá robot navštěvovat, nebo nad kterými nemá uživatel přílišnou kontrolu (například různé diskuzní fóra apod.). Aby se zbytečně nesnižovalo hodnocení stránky vyhledávačem kvůli nekvalitních odkazů. Takovýto odkaz vypadá následovně: `<a href="odkaz" rel="nofollow">text</a>`.

Ovšem co se týče vhodnosti tohoto způsobu, není nikterak lepší než blokování stránek pomocí `meta robots`. Crawler sice nebude následovat daný odkaz, případně nepředá hodnotu odkazující stránky, ale stránku smí nadále indexovat, v případě, že se odkaz nachází kdekoliv jinde na internetu. Dalším negativem je fakt, že ne všechny vyhledávače atribut rel respektují.

Robots.txt

[15] Jde o jednoduchý mechanizmu jak předávat robotům informace o tom, kam na webu smí a kam ne. Navíc může být přesně specifikováno, pro který crawler dané pravidlo platí. Odborně se to nazývá Robot Exclusion Protocol (REP). Jedná se o textový soubor, který se nachází v kořenovém adresáři domény druhého nebo třetího řádu. Název souboru musí být psán malými písmeny. Přístup k souboru je například pomocí URL adresy *http://www.domena.cz/robots.txt*. Hlavní roli v tomto souboru hraje hlavička `User-agent`, kdy lze pomocí proměnných `Disallow` a `Allow` crawlerům definovat co mají na stránkách vynechat a co ne.

Crawleři, kteří navštíví danou doménu, se nejprve podívají, zda na doméně existuje soubor robots.txt. Pokud ano drží se v něm definovaných pravidel. Pokud soubor robots.txt na webu neexistuje je crawler vítán bez omezení. Soubor robots.txt nepodléhá dědičnosti! To znamená, že doména *http://subdomena.domena.cz* nebude podléhat omezením domény *http://domena.cz/robots.txt*. To stejné platí pro dvě domény *http://www.domena.cz* a *http://domena.cz*, pokud jsou z nějakého důvodu odlišné, jejich soubory robots.txt se nebudou vzájemně ovlivňovat. Totéž platí pro protokoly HTTP a HTTPS, avšak většina crawlerů stránky s protokolem HTTPS stejně neindexuje.

Jak na zápis pravidel do souboru robots.txt

Pravidla jsou do souboru zapisována po řádcích. Řádek začínající hlavičkou `User-agent` definuje pro kterého crawlera se následující pravidla vztahují (pokud je uvedena hvězdička, pravidla vztahují na všechny roboty). Následují řádky začínající buďto hlavičkou `Disallow`, která definuje kde crawler nesmí, nebo hlavičkou `Allow`, která naopak definuje kde se crawler může pohybovat.

Crawlerům tak může být zakázán přístup do adresáře *adresář1* a následně povolen přístup do *adresář2*, který se nachází v *adresáři1*. Robots.txt také povoluje dva regulární výrazy a to * (libovolný řetězec znaků) a $ (konec URL adresy). Následují příklady pro objasnění zápisu pravidel:

- `User-agent: *`
  `Disallow: /`
  Zakáže přístup všem robotům na celý web
- `User-agent: Googlebot`
  `Disallow: /*.pdf$`
  Zakázání přístupů ke všem pdf souborům pro crawlera od Googlu.
- `User-agent: SeznamBot`
  `Disallow: /`
  `Allow: /informace/`
  `Disallow: /informace/no`
  Povolení přístupu pouze do adresáře *informace* pro crawlera od Seznamu, s tím, že navíc nesmí načítat stránky, které jsou v adresáři *informace* a začínají řetězcem *no*.

2.4 Indexer

2.4.1 Představení

Co je to vlastně Indexer? Indexer je komponentou vyhledávače, která formuje databázi pro vyhledávací funkce. Ukládá celý text každé stránky nalezené a stažené crawlerem. Každé slovo je uloženo v databázi slov, kde obsahuje informaci, ze kterého dokumentu slovo pochází, kde se v dokumentu nachází a které slova se nacházejí v blízkosti tohoto slova.

2.4.2 Parser

Parsování dokumentu vybere slova dokumentu pro vložení do forward a inverzního indexu. Nalezené slova se nazývají „tokeny“. Proto tedy je parsování k nalezení také pod pojmem tokenizace. Jako první, co většinou každý vyhledávač provede je, že ze získaného textu odstraní tzv. stop slova, to jsou slova jako například spojky, předložky, apod. V češtině se jedná například o slova: a, ale, v, s, atd. Parser také nezajímá veškerý text v dokumentu, ale jen ten, co vidí běžný uživatel, tedy odstavce, odkazy, zvýrazněný text apod. Nezajímají jej například komentáře, které jsou sice součástí zdrojového kódu, ale uživatel je normálně na stránce neuvidí. Z takto získaného a očištěného textu následně může parser přejít ke zmiňované tokenizaci.

Tokenizace

Na rozdíl od lidí, počítače nerozumí struktuře přirozeného jazyka a nemůžou tak automaticky rozeznávat věty a slova. Dokument pro počítač představuje pouze sekvenci bajtů. Neví, že mezery rozdělují slova v dokumentu. Proto musí programátor vytvořit program, jenž se řídí pravidly přirozeného jazyka pro rozlišování slov a vět. Takovéto programy jsou nazývány „tokenizery“, či parsery.

Parser během tokenizace identifikuje sekvenci znaků reprezentující slova a ostatní prvky psaného přirozeného jazyka jako například interpunkci. Parser by měl také zvládat identifikovat entity jako například emailové adresy, telefonní čísla, URL adresy, apod. Při ukládání tokenů můžou být ukládány i jiné informace nežli jen samotné slovo. Například velikost písmen ve slově, jazyk, kódování, slovní druh, pozice v dokumentu, ve větě, pořadí věty v dokumentu, délka, číslo řádku, apod. [9]

Výzvy tokenizace přirozeného jazyka

Nejednoznačnost ohraničení slov a vět

U většiny evropských jazyků, kterých je součástí také čeština, se může zdát úkol celkem snadný a přímočarý, jelikož slova jsou v drtivé většině případů oddělena mezerou. Ovšem problém nastává v případě, že se vyhledávač dostane například na čínské, japonské či arabské stránky, kde již slova nejsou jednoznačně oddělena mezerou. Hlavním úkolem tokenizace je identifikace slov, které bude uživatel hledat. Existuje tzv. specifikace logiky jazyka, která napomáhá v identifikaci slov, resp. jejich oddělovačů, což je nezbytnou součástí v návrhu kvalitního parseru pro jakýkoliv jazyk.

Nejednoznačné jazyky

Vyhledávače ukládají informaci o jazyku či slovním druhu u každého slova, pro správné pochopení a ohodnocení dokumentů. Ovšem získání těchto informací závisí na konkrétním jazyce. Problém je, že ne všechny dokumenty správně identifikují jejich jazyk, či jej neidentifikují vůbec, proto se vyhledávače nemohou spoléhat pouze na tuto informaci a přišli s vlastním řešením, jímž je identifikace jazyka dokumentu.

Rozdílný formát souboru

Aby bylo možné správně rozhodnout, které bajty v dokumentu reprezentují znaky, musíme přesně znát kódování souboru. Může nastat problém, kdy není v dokumentu uvedeno kódování, či je dokonce uvedeno špatně.

Vadné uložení

Kvalita dokumentů není vždy dokonalá, může se například stát, že se v něm vyskytují chyby, ať už chyby přirozeného jazyka či binárního kódování znaků Taktéž může dojít k chybám při přenosu. Je proto nezbytné tyto chyby včas identifikovat a odstranit, jinak se bude docházet ke snižování kvality indexu.

2.4.3 Indexování dokumentů do barelů a seřazení

Poté co je každý dokument rozparsován, je převeden do určitého počtu barelů. Každé slovo je převedeno na wordID s pomocí hash tabulky uložené v paměti, tzv. lexikon. Jamile jsou slova převedeny na wordID, jejich výskyt je zaznamenán do hit listů. Poté jsou tyto informace zapsány do tzv. forward barelů. [9]

Pro vygenerování inverzního indexu, sorter postupně prochází všechny forward barely, které seřazuje podle wordID, čímž vytvoří inverzní barely.

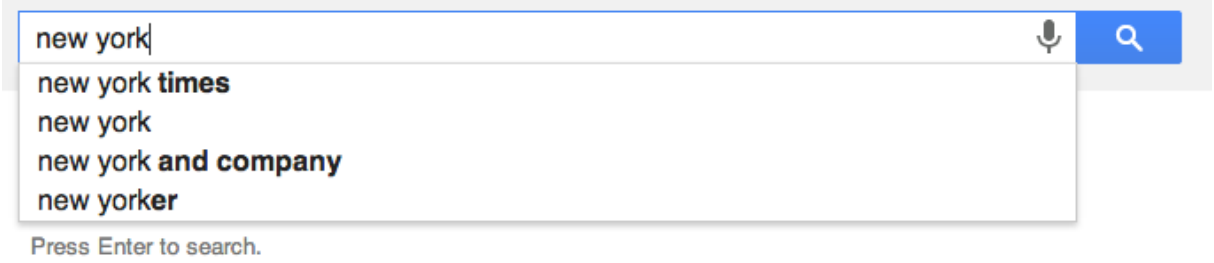
2.5 Query procesor

Query procesor se skládá ze tří částí. První částí je uživateli přístupná stránka pro zadávání dotazů do vyhledávacího pole. Dále je zde program, který vyhodnocuje a zpracovává dotazy a porovnává je s dokumenty uloženými v indexu. A poslední součástí query procesoru je webová aplikace pro formátování a zobrazení výsledků vyhledávání.

2.5.1 Zadání dotazu

Google Suggest

Při zadávání vyhledávacího dotazu napomáhá tzv. našeptávač. Jedná se o aplikaci psanou v AJAXu, která nabízí návrhy vyhledávacího dotazu. Ač se to na první pohled nemusí tak jevit, jedná se o velice mocný nástroj, který může přibližně napovědět frekvenci vyhledávání fráze. Neudává sice přesně četnost vyhledávaného slova, ale dá se předpokládat, že slovo, které je v tomto navrhovaném seznamu na prvním místě, bude vyhledávané častěji než slovo na místě druhém. Dokonce tomu není tak dávno, kdy našeptávač od Seznamu vypisoval frekvenci hledaného výrazu u každého z nich. Bohužel, Seznam již tuto informaci v našeptávači neuvádí. K čemu takto získané informace mohou pomoci? Závěr je jednoduchý, při výběru, na které klíčové slovo bude stránka optimalizována, je vhodné zkontrolovat, jak jsou na tom jednotlivé slova právě v našeptávači vyhledávače, pro který jsou stránky optimalizovány.



Obrázek 3: Google Suggest

Existuje také druhá mocná zbraň našeptávače a tou je, že se lidé nechají lehce ovlivnit nabízenými možnostmi. To znamená, že pokud vyhledávač chce, může cíleně směřovat jeho uživatele na vyhledávání konkrétního dotazu a tím je teoretický směřovat na konkrétní stránky. Ovšem tady vyvstává jeden problém, který není radno podceňovat. Tímto problémem mohou být žaloby a různé sankce za nabízení nevhodných výrazů, například vulgárních či rasistických. Další část práce se tedy zaměřuje na to, jak Google přistupuje k filtrování a řazení nabízeného seznamu. [12]

Řazení

I když bylo zmíněno, že Seznam uváděl u hledaných výrazů frekvenci vyhledávání, je nutné podotknout, že seznam není řazený podle frekvence vyhledávání daného slova. Rozhoduje tzv. popularita. Bohužel, toto téma oplývá tajemstvím a vyhledávače nezveřejňují algoritmy, kterými se jejich aplikace pro výpočet popularity hledaného slova řídí.

V každém případě, kromě popularity seznam ovlivňuje také tzv. *freshness layer*. Jde o ovlivnění seznamu aktuálními či náhlými událostmi. Díky náhlému výkyvu v počtu vyhledávání dané fráze Google usoudí, že se něco děje. Avšak zřejmě má na tento seznam mimo jiné také vliv služba *Google news*. Pokud se tedy něco stane, Google se snaží o to, aby uživatelům zjednodušil vyhledávání. Nejedná se však jen o ovlivnění jedné konkrétní fráze, ale i frází s ní spjaté, například jména zúčastněných osob či míst v dané události.

Dalším a nejdůležitějším kritériem, který ovlivňuje nabízený seznam je tzv. *Personalized searches* neboli personalizované hledání. Pokud je uživatel přihlášen ke službě Google, nabízený seznam je ovlivňován výsledky jeho minulých vyhledávání či vyhledávání jeho přátel.

Oprava pravopisu

Google dává také pozor na to, aby dotazy byly zadány pravopisně správně. A to nejen v anglickém jazyce, ale také v jazyce českém. Například při napsaní slova „VáclavHavel“ nám Google automaticky nabídne frázi „Václav Havel“. Stejně je tomu v případě zadání fráze „pravopys“ či „spellyng“.

Geografické cílení

Výsledky našeptávání se také liší podle toho, ze kterého místa je dotaz zadáván. Ať už se jedná o kontinent, stát či region. Rozdíly se objevují také v případě měst. I když je zřejmé že Googlu nebude rozlišovat každou vesnici, tak například v okolí Ostravy při napsání slova „autobazar“ Google nabídne „autobazar Ostrava“, kdežto například v Táboře Google nabídne „autobazar České Budějovice“.

Filtrování výsledků

Nyní již k samotnému filtrování výsledku a tedy jakési ochraně Googlu před možnými problémy. Filtrování jistým způsobem chrání také potencionální poškozené. Na druhou strnu vyvstává otázka, zda zde nedochází ke kontroverzi mezi svobodou slova a etikou a zákony.

Ochrana před nesnášenlivostí

První, v západním světě velice citlivou položkou, je ochrana před nesnášenlivostí, tedy například rasovou či náboženskou. Tohle tedy u Googlu rozhodně neprojde. Takže pokud je například zadána fráze „nemám rád“ či „nesnáším“, tak Google nabídne fráze jako „vánoce“, loučení“, „lidi“. Rozhodně se vyvaruje rasistických či náboženských výrazů. Filtr konkrétně chrání před nesnášenlivostí typu:

- Rasy a etnikum
- Barvy
- Národnosti
- Náboženské vyznání
- Postižení

- Pohlaví a sexuální orientace
- Lidi různého věku
- Veterány

Soudně vymahatelné škody

Google musel v minulosti často čelit žalobám za „nevhodné“ našeptávání a těmto žalobám rozhodně není konec. Tyto žaloby se většinou týkali osob či firem, které se nabídly v seznamu například na slovo „podvodník“. Na druhou stranu, fráze jako například „podvody na aukru“ Google toleruje. Je tedy otázkou, kde je ona pomyslná hranice, či zdali má Google od dané firmy či osoby povolení.

Kontroverze

Mimo nelegálních a poškozujících slovních spojení jsou tu také kontroverzní fráze. Při zadání slov jako například islám, křesťanství či rasismus je našeptávač očividně pročištěn od nevhodných frází. Je ovšem problém poznat, kdy jsou výsledky opravdu zmanipulované a kdy ne.

Ochrana před warezem a obsahem pro dospělé

Google má velký podíl na ilegálním stahování písniček či filmů z internetu. Bez Googlu by mnohdy nebylo možné nalezení „zapadlých“ odkazů na různých warez fórech. Na druhou stranu se Google snaží proti tomuto problému bojovat, takže například při zadání názvu písničky či filmu spolu s názvem konkrétního warez fóra nebo serveru pro šíření ilegálního obsahu nám našeptávání často nic nenabídne.

Stejně je tomu i u frází s obsahem pro dospělé, které Google naprosto ignoruje. Tedy pokud se nejedná o frázi, kterou Google zatím nezná.

2.5.2 Zpracování dotazu

Zpracovávání dotazu sdílí mnoho stejných kroků se zpracováváním stránek, tedy indexací. Dalo by se říci, že čím více kroků zpracování dotazů obsahuje, tím lepší je vracený výsledek. Avšak je zde problém a to sice kontroverze času a požadovaného výkonu pro nalezení dotazu. Návrháři tedy musí volit kompromis mezi rychlostí a kvalitou. Veřejné vyhledávače většinou upřednostňují čas nad kvalitou, vzhledem k tomu, že obsahují nezměrné množství dokumentů. Zpracování dotazů se skládá z následujících kroků. [11]

Tokenizace

Jakmile uživatel zadá vyhledávací dotaz, vyhledávač musí tokenizovat zadaný text. Neboli rozložit výraz do srozumitelných segmentů. Obvykle jde o rozdělení zadaného textu do slov anebo čísel, dle mezer a interpunkcí. Jedná se v podstatě o obdobný krok, který vyhledávač provádí při tokenizaci stránky.

Parsování

Vzhledem k tomu, že vyhledávací dotazy mohou obsahovat speciální výrazy, jako například booleanovské, spojovací, apod. výrazy, tak vyhledávač nejprve musí dotaz rozparsovat do operátorů a samotných výrazů. Těmto operátorům mohou být přiřazeny znaky interpunkce, jako například uvozovky u Googlu při požadování přesného znění a pořadí zadané fráze, nebo také speciální výrazy jako například výrazy „and“ nebo „or“.

V tomhle bodě mohou již vyhledávače vzít získaný seznam vyhledávacích frází a spustit hledání v inverzním indexu. Ve skutečnosti, je tohle bod, kde některé veřejné vyhledávače již započínají samotné vyhledávání.

Odstranění stop slov a výrazů

Některé vyhledávače však zacházejí dále a ze zadaného výrazu odstraní stop slova, obdobně jako je tomu při zpracovávání dokumentů. Avšak je zde jeden rozdíl a to, že seznam stop slov může také obsahovat celé výrazy jako „chtěl bych informace o“ a podobné výrazy. Nicméně, většina vyhledávačů si vynucuje krátké dotazy, což je patrné už jen ze samotné velikosti vyhledávacího pole, takže můžou tento krok vynechávat.

Vytvoření dotazu

Jak bude vytvořený samotný dotaz, záleží na tom, jak je systém provádí porovnávání. Pokud je používáno statistického porovnávání, tak systém musí obsahovat statistickou reprezentaci výskytu výrazů v dokumentech uložených v systému. Kvalitní systém bude navíc také obsahovat statistickou reprezentaci synonym, aby byl co možná nejvíce relevantní. Je zde další možnost, jíž je booleanovské porovnávání, kdy systém musí vytvořit dotaz obsahující výrazy spojené logickými spojkami and, or nebo not.

Jedná se o další bod, kdy již vyhledávače mohou vzít reprezentaci dotazu a porovnat ji s inverzním indexem. Avšak propracovanější vyhledávače zahrnují ještě další dva kroky.

Rozšíření dotazu

Vzhledem k tomu, že uživatelé obvykle zadávají jen jediný výraz toho, co vlastně hledají, je velice pravděpodobné že vyhledávač rozšíří hledaný výraz o synonyma, čímž se zvýší pravděpodobnost nalezení daného výrazu v dokumentech obsažených v indexu. Proto sofistikovanější vyhledávače rozšiřují výraz o synonyma a dále pak ještě o delší či kratší slovní verze hledaného výrazu.

Ohodnocení dotazu

K ohodnocení dotazu dochází pouze za předpokladu, že dotaz obsahuje více nežli jedno slovo. Poslední krok při zpracování dotazu je ohodnocení významnosti jednotlivých slov v zadaném dotazu. Někdy je tato možnost poskytnuta přímo uživateli, kdy si může určit význam jednotlivých slov v zadávaném dotazu, nebo jednoduše které slova se vy výsledku nacházet musejí a které ne.

Avšak ponechání určení významnosti na uživateli není příliš běžné, jelikož výzkumy prokázaly, že uživatelé nejsou většinou schopni kvalitního ohodnocení. Je zde několik důvodů, proč uživatelé nemohou provádět kvalitní ohodnocení. Zaprvé neví, jaké výrazy se nacházejí v databázi, a výrazy

obsažené v databázi jsou ohodnoceny porovnáním se všemi výrazy v databázi. Zadruhé, většina uživatelů informace, o kterých zatím nic netuší, nemůže tedy logicky ohodnotit významnost slov správně.

Některé vyhledávače mají sofistikované algoritmy pro určení významnosti slov v dotazu, avšak většina vyhledávačů se řídí převážně pravidlem, že první slovo v dotazu je to nejvýznamnější.

Na konci tohoto posledního kroku již i ty propracovanější vyhledávače spouštějí samotné vyhledávání v inverzním indexu.

2.5.3 Vyhledávání a porovnávání

Existuje více než jeden způsob jak najít a porovnávat hledaný výraz vůči dokumentům obsaženým v indexu. Avšak popis všech těchto algoritmů je mimo rozsah této práce, proto se práce jen krátce zmíní o nejznámějších metodách vyhledávání, neboli tzv. „matchingu“. [11]

Vyhledávání v inverzním souboru, ze kterého vyhledávač získá informace, ve kterých dokumentech se hledaný výraz nachází, je prováděno binárním vyhledáváním a je součástí všech vyhledávačů. Nehledě na to, zda své vyhledávání započnou po kroku dva, čtyři nebo pět při zpracovávání dotazu.

Zjištění které dokumenty odpovídají hledanému výrazu, není tedy příliš náročné, alespoň co se z návrhu algoritmu týče. Ovšem s takovýmto výsledkem by se uživatelé rozhodně nespokojili. Proto musejí následně vyhledávače přejít k dalšímu kroku, jímž je výpočet jakéhosi skóre pro jednotlivé stránky. Zde se vypočítávají například podobnosti jednotlivých dokumentů, procentuální zastoupení hledaného výrazu v dokumentu, které se musí pohybovat v určitém, veřejnosti neznámém, rozsahu, jelikož pokud tomu tak není, může být stránka považována za „umělou“. Je zde zohledňována ona významnost jednotlivých slov v dotazu, jak bylo popisováno výše. Dále se zde také počítá s tzv. PageRankem, který bude podrobněji probrán v dalších částech práce. Stručně řečeno, kritérií je mnoho, určitě stovky a zaměřit se na všechny kritéria by bylo pravděpodobně o rozsahu další práce. Navíc tyto kritéria jsou vyhledávači bedlivě střežena a jen minimum z nich se dostalo na veřejnost. Důvod je jasný, je jím zajištění relevance vyhledávání, tedy neposkytnutí návodu vývojářům, jak vytvořit úspěšnou stránku pro konkrétní vyhledávač.

Po vypočítání skóre jednotlivých stránek vrací systém uživateli seznam, seřazený dle jím předpokládané relevantnosti výsledků. Popisem tohoto seznamu se práce zabývá v následující kapitole.

2.5.4 Vracení výsledku vyhledávání

Co přesně stojí za informacemi poskytnutými Googlem a kde tyto informace bere? Zodpovězení této otázky může napomoci při přilákání potencionálního návštěvníka na stránku. [6]

- **Statistický panel** je prvním údajem ve výsledku vyhledávání. Obsahuje přibližný počet nalezených stránek a čas zpracování dotazu, což může napomoci v analýze konkurence.

- Pokud se v dotazu vyskytuje **gramatická chyba**, Google automatický vyhledá informace s opraveným výrazem. Avšak tuto informaci poskytne s nabídnutím vyhledávání dotazu obsahujícího chybu, jelikož chyba mohla být záměrná.
- Občas Google jako další položku zobrazí **nápovědu**, kde může přesměrovávat na stránku s vyhledáváním obrázků, či vyhledávání v jiném jazyce, apod.
- Na řadu přichází samotný **výsledek vyhledávání**, neboli nalezené stránky a jejich stručná specifikace. Seznam těchto stránek je seřazený dle předem vypočítané relevance neboli skóre jednotlivých stránek. K samotným prvkům specifikace stránky:
    - Modrým textem je zobrazený **titulek stránky**, pokud stránka titulek má. Pokud stránek neobsahuje titulek, tak se na tomto místě nachází URL adresa stránky. Po kliknutí na tento prvek je uživatel přesměrován na konkrétní URL adresu.
      Každá stránka by tedy rozhodně měla obsahovat titulek, který je navíc krátký a výstižný a pokud možno titulek, který dokáže zaujmout potencionálního návštěvníka.
    - Každý výsledek vyhledávání obvykle obsahuje **úryvky** ze stránky, které jsou vypsány černým fontem a hledané klíčové slova jsou v nich vyznačeny tučně. Těchto úryvků může být hned několik, každý z nich je potom oddělený třemi tečkami (…).
      Úryvky mohou rovnou poskytnout informace o hledané informaci, co je obsahem dané stránky, nebo můžou vnuknout nápad, jak pozměnit vyhledávací dotaz, aby byly nalezeny tížené informace.
      Může nastat případ, kdy stránku nenavštívil crawler, například z důvodu špatného obsahu, který je obtížný na zpracování nebo z mnohem prostšího důvodu, jímž může být zakázání čtení obsahu. V tomto případě je část s úryvkem vynechána, tedy prázdná.
    - Zeleným textem je vyznačena **URL adresa** stránky s požadovaným obsahem.

2.6 Optimalizace stránek

Již bylo upřesněno, jak fungují vyhledávače a teď se práce bude soustředit na faktory na stránkách a mimo ně, které ovlivňují výsledné hodnocení. První na co se práce zaměří, jsou tzv. off-page faktory, které však budou zmíněny jen okrajově, jelikož tato problematika není hlavním tématem této práce. Značně však ovlivňuje hodnocení stránek z pohledu vyhledávačů. Další části, která bude probrána detailněji, jsou tzv. on-page faktory. Jedná se o jedno z primárních zaměření této práce. Poslední částí této kapitoly bude výběr klíčových slov, který by se teoreticky dal zahrnout do kategorie on-page faktorů. Avšak jelikož se jedná o rozsáhle téma, tak má vlastní kapitolu.

2.6.1 Off-page optimalizace

Co si vlastně pod tímto názvem představit? Stručně řečeno se jedná o faktory ovlivňující hodnocení stránek z vnějšího prostředí internetu. Jedná se tedy především o odkazy směřující na stránku, které mohou, jak bude později probráno, nejen pomoci, ale také uškodit. Jedná se tedy o výměnou zpětných odkazů.

Jaké jsou možnosti?

Zpětné odkazy jdou získávat dvěma základními způsoby. Výměnou anebo koupí. Existují také další možnosti, které budou pro ilustraci uvedeny. Jedná se tedy především o získávání odkazů pomocí registrací do katalogů, účasti ve fórech a diskuzích, na vlastních odlišných doménách, publikování na blozích, na inzertních serverech, apod. Existuje zde také možnost černých praktik, kdy je vybudována síť zpětných odkazů, například pomocí spamů, virů, atd.

Nejvhodnější je však získávání zpětných odkazů přirozenou cestou, což znamená, že je vytvořen tak kvalitní obsah, že uživatelé a ostatní weby budou na stránky odkazovat sami od sebe.

Důležité je si uvědomit, že ve výsledném hodnocení nerozhoduje jen kvantita, ale také kvalita. Zde přichází na řadu tzv. PageRank, který počítá kvalitu stránky právě z odkazů na něj odkazujících.

PageRank

Jedná se o důležitý faktor, který ovlivňuje umístění stránek ve výsledku vyhledávání, není však nejdůležitější. Relevanci stránky Googlu určuje podle asi dvou set různých faktorů. Kolem PageRanku obíhá spousta mýtu. Některé z nich budou hned na začátku vyvráceny, aby se předešlo případným nedorozuměním v dalších částech práce. Mýty se týkají také hodnotících systémů ostatních vyhledávačů. [14]

- *PageRank udává hodnotu jakou naši stránku Google celkově hodnotí.* PageRank udává pouze kvalitu odkazů směřujících na stránku, tedy nejde jen o kvantitu, ale také o kvalitu zpětných odkazů.
- *PageRank je neměnný nebo neklesající.* PageRank je dynamický a může jak klesat, tak stoupat.
- *Čím vyšší je hodnota PageRanku, tím výše bude stránka umístěna ve výsledku vyhledávání.* I když je to částečně pravda, PageRank není jediným a ani nejdůležitějším faktorem při určování pořadí. Může nastat případ, kdy stránka s PageRankem 1 bude ve výsledku vyhledávání výše než stránka s PageRankem 10.
- *Čím více odkazů směřuje na danou stránku, tím větší má stránka PageRank.* PageRanku nezáleží jen na kvantitě, ale také na kvalitě zpětných odkazů.
- *Cílem SEO je vysoký PageRank.* Cílem SEO je také vysoký PageRank, ale mimo to, je zde spousta dalších faktorů, především z části on-page faktorů.
- *PageRank je stejný pro všechny stránky domény.* Každá stránka má svůj vlastní PageRank.

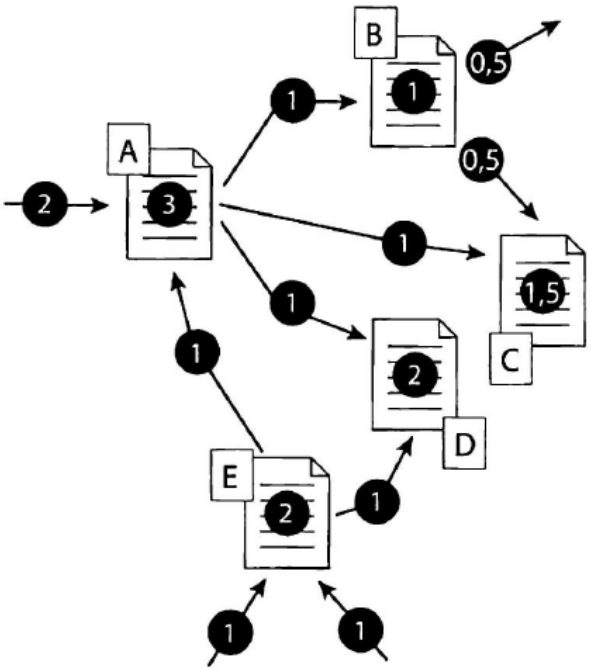
Jak z předchozího odstavce vyplývá, PageRank je hodnotící systém Googlu. Každý vyhledávač má svůj vlastní hodnotící systém, například Seznam má S-Rank. Avšak jak bylo již řečeno, práce se soustředí především na technologie Googlu. Jde tedy o algoritmus Googlu, který na jedenácti stupňové

škále od 0 do 10 ohodnocuje hodnověrnost webové stránky. Autory tohoto algoritmu jsou zakladatelé Googlu Lawrence Page a Sergey Brin. PageRank je pak pojmenovaný podle prvního ze zakladatelů. I když, ač už je to shodou náhod nebo záměrem zakladatelů název PageRank svádí k myšlence, že slovo „page“ v je odvozeno od anglického slova „page“, tedy v češtině „stránka“. Samotný algoritmus vychází z myšlenky Kandall-Weiovy teorie hodnocení, která pochází z padesátých let minulého století a razí ideu porovnávání věcí a lidí na základě vlivu, který na sebe navzájem mají.

Zjednodušená definice z knihy Velký průvodce SEO od autora Michala Kubíčka, která ve stručnosti objasňuje problematiku PageRanku říká:

> *„PageRank představuje hodnotu důvěryhodnosti, tj. kolik stránek současně hodnocených pomocí téhož vzorce na danou stránku odkazuje. Jinými slovy, každá stránka předává část své "hodnověrnosti" stránkám, na které odkazuje.“*

Nyní se práce zaměří na to, jak PageRank ve skutečnosti funguje. Míra předávání hodnověrnosti klesá s počtem odkazů na dané stránce. Odkazuje-li tedy na stránku stránka s PageRankem pět, která obsahuje deset odkazů, předává této stránce větší hodnotu důvěryhodnosti než stránka, která má PageRank šest a odkazuje na sto dalších stránek. Hodnotu PageRanku přitom odkazující stránka neztrácí tím, že obsahuje odkazy, spíše svou hodnotu PageRanku přeposílá nebo předává. Způsob předávání je jednoduše demonstrován na obrázku číslo 4 [1].



Obrázek 4: Princip PageRanku

Výsledkem PageRanku je relativní hodnota, která může v čase jak klesat, tak růst. I když z obrázku může výpočet vypadat relativně jednoduše, jedná se o rovnici s více než pěti sty milióny proměnnými a dvěma miliardami členů.

Co je dobré vědět

Nejprve musí být vytipovány vhodné stránky. Je důležité pamatovat na jednu věc a to, že čím více jsou stránky příbuzné tématu optimalizovaných stránek tím lépe. Algoritmus, který hodnotí příbuznost stránek, se nazývá Topic Rank. Na základě tohoto algoritmu jsou stránky řazeny do tematicky příbuzných clusterů. Je tedy vhodné, pokud má stránka co nejvíce příbuzných stránek z clusteru, ve kterém je sama umístěna.

Důležité je propagovat všechny anebo alespoň co nejvíce stránek optimalizovaného webu. Chybou je, spoléhat se pouze na primární stránku domény. Jako příklad může být uveden jakýkoliv internetový obchod. Je jasné, že pokud by daný obchod propagoval pouze svou úvodní stránku, která většinou obsahuje základní informace o tom, co můžeme v obchodě nalézt, případně nově přidané produkty či různé akce, tak by moc uživatelů na stránky nedorazilo. Jinak tomu bude v případě, pokud internetový obchod bude budovat síť zpětných odkazů pro každý produkt. Rázem se tak začnou na stránky hrnout uživatelé, hledající například konkrétní televizi, či jakýkoliv jiný produkt. [3]

Další velice důležitou věcí, na kterou mnoho webů zapomíná, je správné odkazování na stránky. Jak bylo již zmíněno v popisu indexace stránek, odkazovaný text, tzv. anchor text se přiřazuje stránce, na kterou odkazuje! Je tedy nesmysl, odkazovat na stránky například nápisem „zde“. Text by měl přinejlepším obsahovat název firmy, respektive stránky, a konkrétní produkt či klíčové slovo nacházející se na stránce.

Příklad nevhodného odkazu:

```
Užitečné informace o SEO nalezete <a href=http://domena.cz>zde</a>
```

Správný zápis odkazu:

```
<a href=http://domena.cz>Domena.cz, informace o SEO</a> obsahuje spoustu užitečných informací.
```

Nejlepší zápis odkazu:

```
<a href=http://domena.cz>Domena.cz</a> detailně informuje o problematice <a href=http://domena.cz/seo-problematika>SEO</a>
```

Google bomby

Síla zpětných odkazů může být demonstrována u tzv. Google bomb. Jedná se o snahu odkazovat na konkrétní stránku pomocí fráze, která se na dané stránce vůbec nenachází. Ve většině případů přitom dochází bez vědomí majitele dané stránky. Nejčastější případy bývají například u politiků, kdy se skupina lidí, respektive stránek, rozhodne odkazovat na politika ne příliš lichotivou frází. Následně při zadání této fráze do vyhledávače, vyhledávač vrátí jako výsledek osobní stránky politika. Pro demonstraci stačí zadat do Googlu výše zmiňovanou frázi „zde“. Hned na prvním místě se objeví stránky mapy.cz, které očividně výraz „zde“ nikde neobsahují, ale lidé rádi odkazují na pozici, kde je možno je naleznout, frází „můžete nás nalézt *zde*“. [2]

Google se však snaží proti takovýmto odkazům bojovat. I když někdy může takováto bomba působit jako dobrá reklama, obzvlášť pro zastánce filosofie „lepší je špatná, nežli žádná reklama“. Někdy se

dokonce opravdu povedená bomba dostane do titulků zpráv. Nicméně Googlebot si dává pozor, pokud narazí na přespříliš odkazů, odkazující na stránku výrazem, který se na stránce nachází. Pokud takovýchto odkazů zaznamenal příliš, nedává jim velkou váhu.

Naproti tomu Seznam je na bomby imunní, jelikož zobrazuje pouze stránky, které obsahují hledané klíčové slovo, buďto v textu, titulku nebo URL adresy stránky.

Koupě odkazů

Koupě odkazů je relativně pohodlnou avšak nákladnou metodou získávání zpětných odkazů. Existuje spousta specializovaných serverů, zabývající se právě distribucí odkazů. Jedná se o tzv. PPC systémy, tedy se zkratkou z anglického výrazu „pay per click“, neboli v češtině „platba za klik“. Co se Googlu týče, nabízí se zde systém Google AdWords. Před nákupem takovéhoto odkazu je dobré si zjistit návštěvnost této stránky a také zhodnotit kdo tuto stránku bude navštěvovat, aby nedocházelo k nákupu odkazů na stránce pro automobilové nadšence, pro stránky prodávající domácí elektro. [4]

Výměna odkazů

Výměna odkazů je vedle registrace do katalogů nejsnadnější způsobem získávání zpětných odkazů. Zároveň jedním z nejstarších. Mezi velice populární metody již od prvního rozmachu internetu patří tzv. výměna ikonek. Kdy se jedná většinou o animace či obrázky ne větších rozměrů než 100x40 obrazových bodů, které mají za cíl upoutat a získat nové návštěvníky. Procento jejich úspěšnosti však není příliš vysoké. Přínosem je, že crawler ve zdrojovém kódu nalezne odkaz na vaše webové stránky, nezbyde mu tedy nic jiného, než je navštívit. Nicméně, jak bylo zmíněno, textová část odkazu je velice důležitá, a obrázkový odkaz tuto část postrádá (pomineme-li alternativní text obrázku). [4]

Postupem času se ikony staly pro uživatele v podstatě neviditelnými a v dnešní době je téměř ignorují. Pro tuto situaci existuje dokonce i vlastní výraz, tzv. *bannerová slepota*. V dnešní době se tedy spíše uplatňuje textová výměna odkazů, jelikož pokud nastane případ, že uživatelé na daný odkaz nekliknou, alespoň si zvýšíme hodnocení u vyhledávače. Samozřejmě v případě, že na svou stránku odkazujeme tím správným textem a ze stránky, která má solidní PageRank.

Obecně existují dvě možnosti výměny odkazů:

- První možností je, kdy webová stránka obsahuje podstránku například s názvem *odkazy*, *partnerské weby*, apod., kde jsou umístěny odkazy stránek, které jsou ochotny provést výměnu odkazů s daným webem. Nesmíme zapomenout, aby se odkaz na tuto stránku nacházel na celém webu.
- Druhým způsobem, pravděpodobně kvalitnějším, ale omezený kvantitou je, že do patičky webu jsou umístěny odkazy stránek, se kterými jsme provedli výměnu. Je pravděpodobné, že tato patička se bude vyskytovat na všech podstránkách webu, tedy i na hlavní stránce, která má většinou vyšší PageRank než ostatní stránky. Díky tomu, je zajištěno větší množství odkazů, protože se odkazy nenachází jen na jedné stránce, ale například na dvaceti stránkách. Nedostatkem této metody je fakt, že je omezena počtem takto uvedených odkazů. Správce webu pravděpodobně nebude chtít, aby obsah jeho stránek z devadesáti procent tvořily odkazy.

Pro usnadnění práce s vyhledáváním vhodných webových stránek pro umístění zpětných odkazů existují specializované nástroje. Tyto nástroje můžeme většinou nalézt pod klíčovým slovem „link exchange tool“.

Registrování do katalogů

Další možností výměny odkazů je registrace do katalogů. Nutná je však registrace do správných katalogů, které jsou kladně hodnoceny z pohledu vyhledávačů. Jedná se o katalogy, které vyhledávače považují za relevantní. Zároveň zde platí obdobné pravidlo jako u výměny zpětných odkazů, čímž je správně formulovaný text popisující naši stránku.

Při optimalizaci je dobré nepodceňovat významnost katalogů, jelikož většina crawlerů navštěvuje jako první právě katalogy. Odtud načerpá odkazy a z některý opravdu důvěryhodných katalogů dokonce i popisy stránek. Navíc, to že některé katalogy jsou pro vyhledávače opravdu kvalitním zdrojem informací, napovídá fakt, že se při vyhledávání konkrétního výrazu se nacházejí na prvních místech ve výsledcích vyhledávání právě katalogy.

Definice katalogu

Katalog je web rozdělený dle kategorií a obsahující odkazy na jiné stránky. Vyhledávací funkce v katalozích jsou dnes již samozřejmostí a katalogy tak vyhledávají ve své databázi titulků, popisu stránek či klíčových slov stránek, pokud požaduji při registrování stránky definici klíčových slov na nich obsažených.

Problémem katalogů je fakt, že při registraci si uživatel definuje titulek, popis či klíčové slova stránky. Pokud zde neexistuje administrátor, který by všechny nově registrované stránky procházel a kontroloval, zda zadané údaje souhlasí s údaji zadanými při registraci do katalogu, můžou se tyto informace stát irelevantními.

Mezi nejznámější světový katalog patří Yeahoo!. U nás je to katalog od Seznamu, Atlasu a Centra. Nejkvalitnějším katalogem na světě je však katalog ODP, z anglického názvu „open directory project“, nacházející se na adrese *http://dmoz.org*. Tento katalog upravují a kontrolují dobrovolníci po celém světě. Registrace stránek právě v tomto katalogu je z hlediska SEO velice důležitá, jelikož právě kvůli nestrannosti a relevantnosti tohoto katalogu jsou odkazy v tomto katalogu vyhledávači kladně hodnoceny a považovány za důvěryhodné. Google dokonce považuje ODP za takovou autoritu, že pokud o daných stránkách nenajde dostatek informací na stránce samotné, zobrazí ve výsledcích vyhledávání popisek či titulek právě z tohoto katalogu.

Správné využití katalogů

Pro to, aby katalogy byly pro stránky přínosem, musí být dodrženo pár základních pravidel:

- Pokud to daný katalog umožňuje, měl by titulek obsahovat nejdůležitější klíčové slova a slovní spojení.
- Text pro popis stránky je většinou krátký, okolo dvě stě padesáti znaků. Je dobré pokusit se o zajímavé zachycení obsahu stránek, které by bylo případně schopně upoutat potencionální návštěvníky. Samozřejmostí je, aby tento text obsahoval klíčové slova stránky.
- Pokud katalog umožňuje zaregistrovat odkazy na podstránky daného webu, je vhodné pro každou stránku vymyslet vlastní text a titulek s klíčovými slovy na této stránce obsaženými.
- Posledním tipem je, založit novou emailovou schránku, či dočasnou emailovou schránku, která je využita pouze pro potvrzení registrace do katalogu. Důvodem je, že katalogy často využívají svou činnost i k jiným marketingovým účelům a mohlo by tak lehce dojít k zahlcení hlavní emailové schránky nevyžádanou poštou.

Jak zjistit odkazující stránky

Pro zjištění, které stránky odkazují na konkrétní stránku, existuje speciální příkaz, který se zadává rovnou do políčka pro vyhledávání. Jde o dotaz ve tvaru *link:www.domena.cz*. Vyhledávač vrátí seznam stránek odkazující na zadanou stránku. Co se Googlu a Seznamu týče, tyto dva vyhledávače příkaz link nepodporují. Stačí však zadat adresu do políčka pro vyhledávání přímo, bez prefixu *link*, a vyhledávač vrátí seznam stránek, které odkazují na zadanou URL adresu. Případně existují specializované zahraniční aplikace plnící funkci informování o zpětných odkazech.

2.6.2 On-page optimalizace

On-page optimalizace je oproti off-page optimalizaci zaměřená na zdrojový HTML kód stránky a její textový obsah. Jedná se tedy především o validitu kódu, s důrazem na nevynechání některých HTML elementů a atributů, a teorii tvorby a výběru klíčových slov na stránce. Existuje pár základních zásad, na které by se při tvorbě stránek nemělo zapomínat:

- Je vhodné, když je HTML kód stránky validní. Všechny elementy v dokumentu by měly být uzavřené, předejdeme tomu tak situacím, kdy crawler špatně rozparsuje stránku.
- Na stránce by se neměly objevovat nefunkční odkazy, tedy odkazy vedoucí na neexistující stránky. Pokud se na stránce takovéto odkazy nacházejí, tak crawleři nemusejí být schopní efektivně stránky indexovat a může se stát, že se stránky neobjeví ve výsledcích vyhledávání.
- Dáváme přednost statickým URL adresám před dynamickými, tzv. SEO friendly.
- Zajištění kvalitního obsahu. Tomuto tématu se budeme podrobněji věnovat v následujících částech práce.
- Text stránky musí obsahovat klíčové slova, na které chceme být vyhledáváni.
- Dodržování rozumné velikosti stránek. HTML kód stránky spolu s textem by neměl přesahovat 100 kB. Podaří-li se velikost stránky udržet pod 40 kB, bude to ještě lepší.
- Každá stránka musí být dostupná alespoň z jednoho odkazu.

- Text, který má být zaindexován, musí být umístěn mimo obrázky. Pokud je název firmy uveden například pouze jako obrázkové logo v hlavičce stránky, uživatelé tuto stránku pod názvem firmy nenajdou.

Klíčová slova

Tento úkol není radno podceňovat, jelikož správně zvolená klíčová slova jsou opravdu klíčová pro umístění stránek ve výsledku vyhledávání. [1] [2] [3] [5]

Důležitým faktorem při výběru vhodného klíčového slova, na které bude stránka optimalizována je **relevantnost** daného klíčového slova. Je nepřijatelné stránky autobazaru optimalizovat na klíčové slovo „cukrářství“. Většina laiků si řekne, že nejlepší bude optimalizovat autobazar na slovo „autobazar“. Ovšem tato volba není tak jednoznačná, jelikož je-li bráno v potaz, že se daný autobazar nachází například v Ostravě, tak při frekvenci vyhledávání slova „autobazar“ tisíc krát denně se na danou stránku podívá v ideálním případě tisíc lidí, ale nezdrží se zde déle než pět sekund a naštvaně opustí stránku, jelikož lidí co hledají zrovna autobazar v Ostravě, může být například deset. Je tedy dobré, co nejpřesněji a nejrelevantněji zvolit ono klíčové slovo, pro omezení plýtvání úsilím a finančními prostředky na něco, co se v budoucnu nevrátí. V případě zmiňovaného příkladu autobazaru z Ostravy, by bylo ideálně zvolené klíčové slovo „autobazar Ostrava“. Frekvence vyhledávání tohoto slovního spojení sice nebude tak vysoká jako samotného slova „autobazar“, avšak do budoucna přinese mnohem větší užitek, jelikož stránky jsou cíleny na uživatele z okolí Ostravy.

Dalším faktorem, nad kterým je potřeba se při výběru zamyslet je tzv. **long tail**. Jde o přirovnání potencionálních zákazníku nebo jen návštěvníků stránek ke kometě, přičemž hlavní myšlenkou je fakt, že kometa se neskládá pouze z hlavního proudu (head), ale také velkým počtem malých segmentů, neboli ocas komety (tail). Jde o to, že množství potencionálních návštěvníků v „ocasní části“ může být mnohem větší, než pokud zacílíme naše soustředění pouze na hlavní proud.

Právě ve strategii long tail se skrývá tajemství úspěchu většiny internetových obchodů, jelikož oproti kamenným obchodům si nemusí udržovat sklady a nemusí se bát, že nakoupí větší množství produktů než je skutečná poptávka. Mohou mít tedy nabídko mnohonásobně větší a tím uspokojit větší množství zákazníků, i když se nesoustředí na hlavní proud trhu. Nakonec mají větší zisk z menšího prodeje, ale větší nabídky, nežli naopak.

Podobně je tomu i u klíčových slov. Například v případě internetových obchodů existují různé fáze nákupu. Například uživatel, který se rozhodne koupit si nový fotoaparát, bude na začátku vyhledávat „digitální fotoaparát“, poté specifikuje vyhledávání například o položky velikosti čipu a zoomu. Nakonec si pravděpodobně vytipuje pár produktů, které bude porovnávat, bude tedy vyhledávat konkrétní produkty. Je na každé stránce, na kterou tuto část se zaměří a bude se tak snažit získat potencionální zákazníky.

Co by mezi klíčovými slovy nemělo chybět

Při vymýšlení nejvhodnějšího klíčového slova pro optimalizované stránky nesmí být opomenuta jedna velice důležitá věc, kterou je název společnosti. Někdo by mohl namítat, že to je logické, pravdou však je, že ne jedna firma nejde nalézt pod svým vlastním názvem.

Dalším možným zvážením jsou například překlepy. Často se stává, že uživatel napíše klíčové slovo špatně a nalezne zcela něco jiného, než původně chtěl, je tedy třeba zamyslet se nad tím, zda taky tyto překlepy nezahrnout do klíčových slov, obzvlášť jedná-li se o cizí slovo. Dokonce existují techniky, které jsou postavené právě na tom, že stránka počítá s návštěvníky, kteří je vyhledali omylem, například při špatném zadání názvu.

Při výběru se musí soustředit na to, jak bude potencionální návštěvník hledat. Místo soustředění se na to, zda je klíčové slovo psané odborně a korektně. Například, pokud se bude jednat o internetový obchod prodávající chladničky, který bude optimalizovaný na klíčové slovo „chladnička“, pak se pravděpodobně velkého přísunu zákazníků nedočká, protože 90% laiků nenazývá chladničku chladničkou, nýbrž ledničkou či lednicí.

Mělo by být pevně určeno tři až pět klíčových slov pohybujících se napříč celým webem, které budou doplňovány klíčovými slovy specifickými pro každou stránku.

Jak vybírat klíčové slova

Existují stránky, které zobrazí frekvenci vyhledávání klíčových slov. Většinou se jedná o stránky přidružené k samotným vyhledávačům, které takto poskytují silný nástroj při rozhodování na které klíčové slovo stránku optimalizovat. Google nabízí takovýto nástroj v souboru nástrojů pod záštitou Google AdWords. Není tomu dávno co Google upustil z klasického udávání statistických hodnot k tzv. plánovači klíčových slov, který je sám schopný zanalyzovat stránku či nabídnout vhodné slova pro danou orientaci firmy či produktu, spolu s předpokládanou cenou klíčového slova v případě investice do systémů PPC. Je třeba zmínit, že statistiky o prokliknutí jsou nejpřesnější právě ze systémů PPC. I když zde vyvstává problém konkurenčních firem, které úmyslně proklikávají odkazy, aby donutili konkurenční firmu bezvýsledně platit.

Důležitou věc, která musí být při výběru klíčového slova zvážena, je nejen frekvence vyhledávání, ale také síla konkurence na dané klíčové slovo. Mnohdy může být mnohem efektivnější optimalizovat stránky na klíčové slovo, které je vyhledávané pět set krát za den s nízkou konkurencí, než slovo, které je vyhledávané deset tisíc krát za den se silnou konkurencí, jelikož dostat stránky na první příčky v silné konkurenci bude jak časově tak finančně náročné.

Tematická analýza

Tematická analýza klíčových slov je dělena na tzv. *vertikální* a *laterální*. **Vertikální** analýza je rozbor klíčových slov souvisejících s tématem stránky, a jelikož jde napříč oborem, je nazývána vertikální analýzou. Jako příklad může být uveden internetový obchod prodávající hokejové potřeby. Zde budou klíčové slova po vertikální analýze například názvy jednotlivých produktů brusle, hokejka, puk. Budou se zde však nacházet i například názvy jednotlivý postů, například, útočník, brankář, trenér, apod.

Laterální analýza naproti tomu uvažuje do větší šířky a vydává se i mimo samotný hlavní obor. Snaží se nalézt klíčové slova příbuzná ale nepřímo spjatá s předmětem podnikání. V případě hokejového obchodu může potencionální zákazník například hledat školu bruslení či statistiky z minulých utkání a údaje z historie. Internetový obchod tak může získat potencionálního zákazníka při optimalizování na

klíčové slovo „škola bruslení“. Nebo, může vytvořit se stránkami obsahující tyto informace síť zpětných odkazů, jelikož pokud se někdo nachází na stránce zabývající se školou bruslení, pravděpodobně si bude chtít také koupit brusle.

Jak lidé vyhledávají

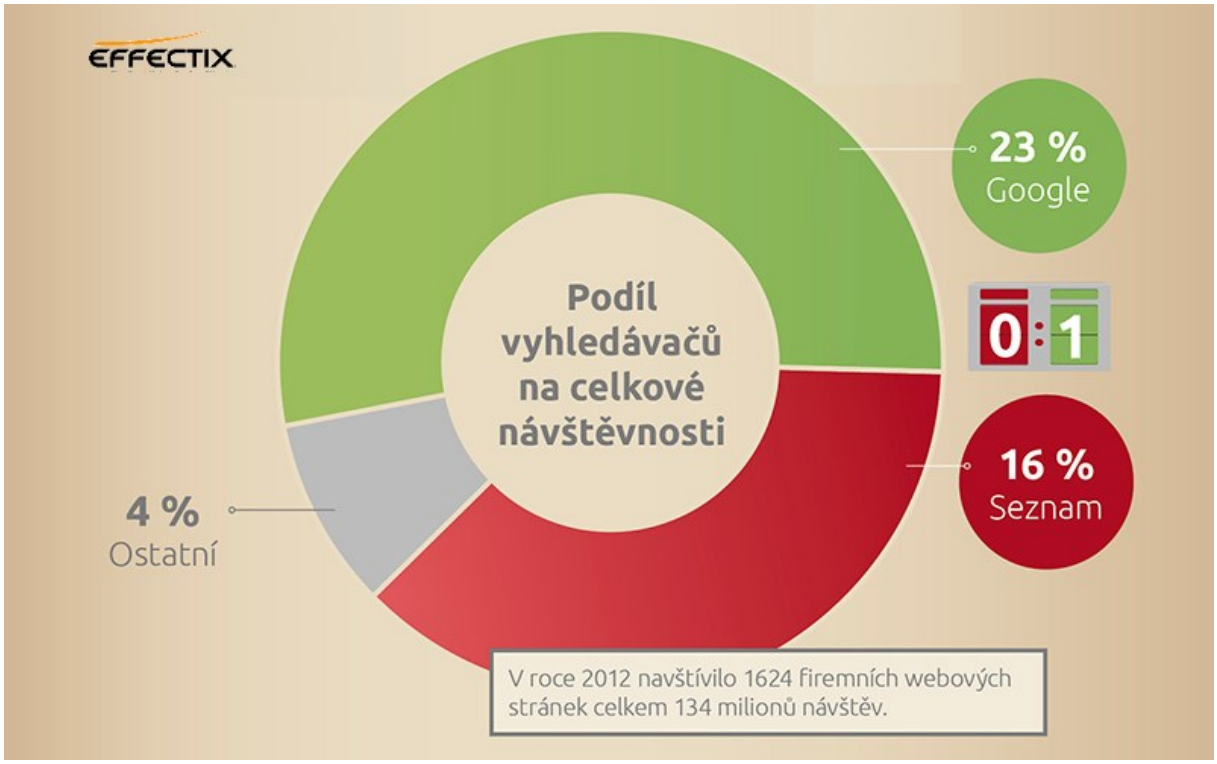
Při výběru klíčových slov musí být zváženo, jak se lidé vyhledávačů dotazují. Je vhodnější optimalizovat stránky na jedno slovo či na frázi? A pokud frázi, tak z kolika slov? Existuje zde dlouhodobá statistická analýza firmy OneStat, která zveřejnila výsledky svého výzkumu [16]:

- 30% lidí hledá dvouslovně
- 28% lidí hledá trojslovně
- 16% lidí hledá čtyřslovně
- 13% lidí hledá jednoslovně
- Zbytek hledá pěti, šesti a sedmi slovně

Je tedy zřejmé, že nejlepší při výběru klíčových slov je zaměřit se na dvouslovné a tříslovné fráze.

Statistiky také poukazují na to, že se lidé neobtěžují s psaním velkých písmen, například na začátku jmen. Ostatně, vyhledávače stejně převádí veškerý text na malé písmena, pro jeho lepší zpracování, takže tímto problémem se trápit nemusíme. Totéž platí o skloňování či jednotném a množném čísle. I když se ukázalo, že většina lidí preferuje při zadávání dotazu množné číslo před jednotným.

Jak již bylo v předchozích částech práce zmíněno, velice také záleží na tom, pro který vyhledávač jsou stránky optimalizovány. Jak je to tedy s podílem vyhledávačů na českém trhu? Společnost Effectix přišla v roce 2013 s výsledky analýzy podílu Googlu a Seznamu na návštěvnosti webových stránek (*http://www.doba-webova.com/)*. Analýza zavedla jakýsi skórovací systém, ze kterého vítězně vzešel Google s poměrem 5:3. Je tedy zde zahrnuto celkem osm analýz obou předních českých vyhledávačů plus analýza celkového podílu vyhledávačů na návštěvnosti webových stránek, ze které vzešlo, že rovných 44% návštěvníku zamíří na stránky právě pomocí vyhledávačů. Což je umístění na první příčce, následují přímé návštěvy s 20% podílem. Nicméně, poměr zdrojů návštěvnosti Googlu vůči Seznamu (včetně relevantních zdrojů u Seznamu, jako je například stránka firmy.cz, zboží.cz, apod.) je 49% k 42% pro Google. Bez relevantních zdrojů je tomu potom 53% k 37% pro Google. Je tedy jasné, že po dlouholeté bitvě je u nás již na špici Google, ostatně tak jak je tomu v celém světě. [13]



Obrázek 5: Podíl vyhledávačů na celkové návštěvnosti

Umístění klíčových slov

Byla probrána problematika klíčových slov, konkrétně jak je vybírat a na koho je soustředit. Nyní se práce zaměří na to, kde ve zdrojovém kódu je vhodné umisťovat klíčové slova. Nejedná se jen o zdrojový kód optimalizované stránky, ale také kód odkazujících stránek, respektive odkazů na těchto stránkách.

Již byl objasněn princip funkčnosti internetových vyhledávačů, takže je zřejmé, že *klíčové slova se na stránkách prostě vyskytovat musí*. Práce se mimo jiné bude také soustředit v jakém množství klíčové slova uvádět.

Kde vidí vyhledávače klíčové slova:

- URL adresa stránky (*www.klicove-slovo.cz, www.domena.cz/klicove-slovo/*)
- Titulek stránky – `title`
- Obsahové metaznačky – `description, keywords`
- Tělo stránky
  - Popisy obrázků – `<img alt="…" />`
  - Nadpisy – `h1, h2, h3, h4, h5, h6`
  - Odstavce – `p`
  - Titulky odkazů – `<a title="…"></a>`
  - Samotný text odkazů, tzv. anchor text - `<a>…</a>`

- Zvýrazněný text – `strong, em, b, i`
- Seznamy – `ul, li`
- Definice – `dd, dt`

Kde vyhledávače klíčové slova ignorují/nevidí:

- Obrázky
- Text generovaný JavaScriptem
- Části stránky generované AJAXem
- Animace, obrázky, videa ve Flashi

Danny Sullivan, šéfredaktor Search Engine Watch:

> *„Webové stránky ve Flashi působí stejně, jako kdybyste se představovali nepopsanou vizitkou.“*

Výše uvedené doporučení nelze brát jako dogma pro každý vyhledávač. Bylo již řečeno, že každý vyhledávač má své vlastní algoritmy na parsování a hodnocení webu, tím pádem je pro něj každý element či atribut jinak významný.

Nesmí být opomenuta znaková sada stránky, která se uvádí v hlavičce. Například Seznam indexuje pouze stránky se znakovou sadou UTF-8, ISO-8859-2 a WINDOWS-1250.

Doména

Ideální doména by měla být taková, která obsahuje ve svém názvu zaměření stránek. Navíc, pokud doména obsahuje přímo klíčové slova, je pravděpodobně již dlouho zabraná a tím pádem považována návštěvníky za relevantnější. Název domény by se zároveň měl volit co nejkratší a dobře zapamatovatelný. Jen těžko si zapamatujeme název složený z padesáti znaků.

Marek Prokop, H1:

> *„Z pohledu SEO mají klíčová slova v doméně v podstatě tři významy: Jako text zpětného odkazu, nejčastěji na úvodní stránku, ale i na podstránky – pak je důležitá kombinovatelnost slov v doméně s dalšími slovy do konkrétnějších frází. Totéž ale na úrovni klíčových slov v URL. A nakonec pro navigační dotazy - pak je důležitá přesná shoda dotazu (případně s vynechanými mezerami) a doménového názvu.“*

Pomlčky

Je lepší mít dvou a více slovnou doménu s pomlčkou nebo bez? Co se z marketingového hlediska týče, lépe se propagují domény bez pomlčky. Stačí, se zamyslet nad reklamou v rádiu. Jak asi zní doména s pomlčkou a bez? Navíc, pořád existují uživatelé internetu, kteří neví, kde se vůbec pomlčka na jejich klávesnici nachází. Měli by být zohledněni všichni potencionální zákazníci.

Z pohledu vyhledávačů je situace odlišná, jelikož pomlčky anebo tečka berou jako oddělující znaky v doméně a chápou tedy text mezi nimi jako samostatné slovo. Ovšem neznamená to, že by doména

bez pomlčky byla ve značné nevýhodě, jelikož při vyhledávání vyhledávače spojují slova domény a slova z dotazu a porovnávají sloučený název domény se sloučeným vyhledávacím dotazem.

Jaké je tedy ideální řešení? Odpověď je jednoduchá, je-li to možné, je nejlepší pokusit se zaregistrovat jak doménu s pomlčkou, tak doménu bez pomlčky. Tímto krokem je zároveň zamezeno konkurenci koupit druhou doménu a přesměrovávat pomocí této domény na svoje stránky.

Co se může stát silným nástrojem, při zjištění, že je požadovaná doména již zabraná, jsou subdomény. Lze tak například vytvořit doménu *http://prodej.bazenu.cz*, či jednoduše *http://prodej-bazenu.mojefirma.cz*.

Stáří domény

Při zamýšlené obměně domény, za doménu obsahující klíčové slovo, je vhodné si tento krok dvakrát rozmyslet, jelikož stáří domény hraje také významnou roli. Například je všeobecně známo, že Google upřednostňuje starší domény před těmi novějšími. Je tedy možné, že nežli zaměnit celou doménu, bude lepší předělat pouze obsah stránek pro dosažení lepších umístění ve výsledcích vyhledávání.

Koncovka domény

Pozor na koncovky. I když se u Googlu nesetkáme s větším problémem u jakékoliv koncovky, tak naopak u Seznamu bude pravděpodobně výše doména s českou koncovkou. Není tomu tak dávno, co Seznam dokonce vůbec neindexoval stránky končící jinou koncovkou nežli .cz. Tedy indexoval, ale jen na přímé požádání, crawler automaticky na doménu s koncovkou .com či .eu nezabloudil.

S nebo bez www na začátku?

Pozor na to, zda je před doménou uváděno www nebo ne. Vyhledávače považují tyto dva způsoby zápisu za dvě odlišné domény. Je jim jedno, která varianta je zvolena, musí být však zajištěno, aby bylo na stránky odkazováno pouze zvolenou variantou. Jako opatření, jelikož nelze zabránit tomu, aby na stránky nikdo neodkazoval druhou variantou, zavést u druhé varianty přesměrování. Zabrání se tak zbytečnému snížení hodnocení kvality webu kvůli duplicitnímu obsahu (vyhledávač si za indexuje do databáze stránky s www i bez www a následně jednu z nich smaže a označí za duplicitní). Bohužel, nad tím, která stránka bude smazána, má kontrolu pouze vyhledávač.

URL adresa

Stejně tak jako je důležité mít klíčové slova v názvu domény, je důležité mít klíčové slova v URL adrese stránky. Pravděpodobně dokonce důležitější, jelikož Google propaguje, že jej klíčové slova v názvu domény vůbec nezajímají. Mnoho lidí může namítat, že je to nesmysl, protože pokud mají doménu s klíčovými slovy, jsou většinou na předních příčkách. Vyvstává zde však otázka. Není náhodou příčinou anchor text v odkazech, které odkazují na stránku právě názvem domény? Tím se zajistí výskytu klíčových slov v anchor textu a stránka jde v hodnocení na dané klíčové slovo rázem nahoru. [3]

Ale zpět k URL adrese. Jak bylo již zmíněno, URL adresa by měla být SEO friendly. Jak vypadá SEO friendly adresa bude nejlépe objasněno na příkladech v tabulce 3.

Běžná URL adresa	SEO friendly URL adresa
http://obchod.cz/index.php?kategorie=2&znacka=11	http://obchod.cz/televize/samsung
http://magazin.cz/doc/index.aspx?id=25&obdobi=5	http://magazin.cz/kveten/relax-v-prirode
http://univerzita.cz/sc.pl?fak=1;kat=3;kod=it21;info=15	http://univerzita.cz/fakulta/katedra/it/rozvrh

Tabulka 3: SEO friendly URL adresy

Jednotlivé stránky se navenek jeví jako pevné dokumenty, nikoliv dynamické stránky s množstvím předávaných parametrů. A pro vyhledávače je stejně jako pro lidi jednoduší si zapamatovat statické nežli dynamické adresy. Pokud se navíc v takovéto adrese vyskytuje nějaké to klíčové slovo, okamžitě si v očích vyhledávače přilepšíme.

Důležité také je, aby adresa byla neměnná. A pokud musí být adresa stránky opravdu změněna, mělo by být zajištěno, aby stará adresa přesměrovávala uživatele a crawlery na novou adresu.

Háčky a čárky v URL adrese

Co se týče diakritiky v URL adrese, je lepší se těmto znakům vyhnout. Problémem je, že URL adresa nemá přesně specifikovanou znakovou sadu a vyhledávače potom neví jak adresu správně parsovat. Existují sice zástupné znaky začínající znakem %, za kterým následuje číslo (například mezera se označuje jako %20), ale tohle není zrovna nejšťastnější řešení, jelikož na čitelnosti to URL adrese rozhodně nedodá.

Duplicita URL adres

Podobně jako je tomu u domén s a bez www na začátku jsou na tom i URL adresy, které mají buďto specifikovanou nebo prázdnou domovskou stránku. Konkrétně jde o případ, kdy je domovská stránka ve tvaru *http://domena.cz/* a ve tvaru *http://domena.cz/index.php*. Tyto dvě stránky, i když jsou fyzicky stejné, jsou opět považovány za odlišné a označeny jako duplicitní. Vyhledávač opět jednu smaže z databáze a stránka tak zbytečně přichází o body. Měli by tak opět být zajištěno přesměrování pouze na jednu variantu.

Titulek stránky

Titulek stránky je z hlediska SEO on-page optimalizace tím nejdůležitějším prvkem. Jak bylo již zmíněno, vyhledávače jej uvádějí jako odkazovaný text ve výsledku vyhledávání. Navíc, pokud titulek obsahuje ty správné klíčové slova a koresponduje s obsahem stránky, může výrazně napomoci ve výsledcích vyhledávání.

Christine Churchill:

> *„Pokud byste měli čas jen na jedno opatření SEO na svém webu, věnujte ho tvorbě dobrých titulků stránek.“*

Nestačí se však soustředit se pouze na to, aby klíčové slovo bylo obsaženo v titulku stránky. Vyhledávače sice neignorují stránky, které mají klíčové slovo jen v titulku, ale v konkurenčním boji se tyto stránky neumístí na přední pozice, pokud neobsahují klíčové slovo také v textu stránky.

Není vhodné, aby byl titulek přecpaný klíčovými slovy nebo zbytečně moc dlouhý. Všeobecně se uznává pravidlo, že by titulek neměl být větší než sedmdesát znaků. Zároveň je taky důležité pořadí slov v titulky, kdy první slovo má nejvyšší váhu, pozor tedy na titulky začínající slovy „web“, či „webová stránka“, zbytečně tak stránku okrádáme o lepší umístění. Další důležitou informací je fakt, že titulek by neměl být na doméně duplicitní, neboli každá stránka by měla mít svůj vlastní titulek.

Deset zásad s klíčovými slovy

1. Klíčové slovo v URL adrese
2. Klíčové slovo v doméně
3. Klíčové slovo v `title` s optimální délkou 10-70 znaků
4. Klíčové slovo ve značce `meta` typu `description` s délkou do 200 znaků
5. Klíčové slovo ve značce `meta` typu `keywords` s počtem slov ne větším než 10
6. Frekvence výskytu všech klíčových slov v těle stránky by se měla pohybovat v rozmezí 5-20%, frekvence individuálního klíčového slova potom v rozmezí 1-6%
7. Klíčové slova v nadpisech `h1`, `h2` a `h3`
8. Klíčová slova zvýrazněná pomocí značek `b`, `u` a `strong`
9. Blízkost klíčových slov – „levné kuchyně“ je vhodnější než „levné kuchyňské linky a kuchyně“
10. Dodržovat pořadí klíčových slov – na pořadí klíčových slov záleží

Navigace a odkazy na stránkách

Existují dvě zásady, kterými je důležité se při tvorbě navigace na webu řídit [1]:

- Navigace je stálá – návštěvník stránek ji najde vždy na svém místě
- Aktuální stránka, na které se návštěvník nachází, je v menu vyznačená

Druhy navigace:

- Základní – klasické menu
- Odkazy v patičce nebo hlavičce
- Doplňková, neboli drobečková
    - Často součástí internetových obchodů, kdy nám udává pozici na které se na webu nacházíme, například: úvodní stránka – podkategorie – produkt
- Mapa stránek (sitemap)
- Vyhledávání
- Odkazy v textu

Chyby v navigaci

Mezi nejčastější chyby v navigaci patří fakt, že často nejde poznat, že se jedná o navigaci. Například obrázková navigace.

Dalším problémem je nekonzistence, kdy se odkazy na každé stránce nacházejí na jiném místě.

Velkou chybou je, pokud navigace na stránkách zcela zmizí a jediný způsob jak se k ní dostat zpět je dostat se na domovskou stránku ať už pomocí URL adresy nebo tlačítka *zpět* v prohlížeči.

Musí se také počítat s tím, že se návštěvník na web nedostane přes úvodní stránku. Je tedy důležité zajistit, aby byl z každé stránky přístup na jakoukoliv jinou stránku.

Odkazy, které odkazují na již otevřenou stránku nebo na samy sebe také nejsou vhodnými. Není vhodné nuceně otevírat stránky v novém okně či panelu, není-li to opravdu nutné.

Chybná z hlediska SEO je navigace vytvořená pomocí JavaScriptu, Javy, AJAXu, Flashe, atd. Takováto navigace je pro vyhledávače neviditelná.

Na každé stránce by měl být odkaz na úvodní stránku případně, pro zpřehlednění, použita drobečková navigace.

Lokální odkazy

Pokud se na stránce vyskytuje opravdu dlouhý text, je dobré na konec tohoto textu umístit odkazy, které zavedou návštěvníka zpět na hlavičku stránky. Pokud se jedná o dlouhý text, který je vhodné rozdělit na odstavce, je dobré před tímto textem vytvořit odkazy na jednotlivé odstavce a zpět z odstavců na odkazy. Tento systém můžeme například vidět v případě wikipedie.

Odkazy v textu

Odkazy v textu by se měli řídit jednou grafickou úpravou pro celý web, nejčastěji jsou uživatele navyklí na barevné rozlišení odkazů, plus podtržení těchto odkazů. Musíme pamatovat například na zrakově postižené lidi, kteří jsou barvoslepí, takže při pouhém barevném zvýraznění odkazu nemusí tento odkaz rozeznat od zbytku textu.

Univerzální mapa stránek pro vyhledávače (sitemap.xml)

Každý crawler ocení a přidá na hodnocení webu, který obsahuje soubor sitemap.xml. Jedná se o výčet všech stránek, které jsou k nalezení na webu, spolu s informací, kdy byla konkrétní stránka naposledy aktualizována či vytvořena a jak častá je frekvence aktualizace této stránky.

Existuje celá řada nástrojů, které vygenerují soubor sitemap.xml. Je zde ovšem podmínka kvalitní navigace na webu, jelikož se tyto aplikace stejně jako crawleři nedostanou tam, kde nevede na webu žádný odkaz.

10 tipů jak pomocí navigace dosáhnout lepšího umístění

1. Interní odkazy obsahující klíčové slova – stejně jako anchor text, i samotné URL by mělo obsahovat klíčové slova, je tedy dobré vhodně volit názvy souborů. Pokud se jedná o víceslovné soubory, oddělíme slova pomlčkami.
2. Zkontrolování, zda jsou všechny interní odkazy správné a funkční.
3. Logická stromová struktura odkazování s ne více jak dvěma či třemi prokliky na každou stránku.
4. Nezapomínat na vnitřní odkazování u obsáhlých stránek.

5. Dávat si pozor, kam odkazujeme, abychom neodkazovali například na spamové stránky či link farmy. Takovéto stránky nás mohou v očích vyhledávače poškodit.
6. Text v odkazu – vyhnout se odkazům typu „zde“, raději zvolit vhodné klíčové slovo.
7. Odkazy by měly být v čase stabilní – pokud odkazují na určitou stránky, neměly by další den odkazovat na jinou.
8. Pravidelná kontrola zda jsou stránky, na které odkazujeme funkční. Týká se to jak vnitřních, tak vnějších. Existuje řada online nástrojů, které kontrolu provedou za nás.
9. Nepřehánět to s extérními odkazy. Google uvádí, že toleruje 100 extérních odkazů na stránku, avšak někdy toleruje i více.
10. Délka odkazů – neměla by překročit více jak 2000 znaků, přičemž ideální je do 100 znaků.

2.6.3 SEO copywriting

Copywriting je odvětví zabývající se psaním poutavého a kvalitního textu. Jakožto takové je zahrnuté také v SEO, jelikož je důležité napsat kvalitní text, neboli potravu pro crawlery, obsahující námi vybrané klíčové slova. [1] [2] [3] [4] [5]

Důležitým faktorem, které může stránky ve výsledcích vyhledávání posunout na výší pozice, jsou aktuální články. Je tedy potřeba neustále aktualizovat stránku a přidávat nové články, ale ne tak ledajaké, ale takové, které budou mít vyhledávače rády. Budou tedy obsahovat přiměřené množství klíčových slov. Studie prokázaly, že tato část SEO je nejdůležitější a dokáže nejvíce ovlivnit umístění webových stránek ve výsledcích vyhledávání.

Častým problémem jsou například internetové obchody, které sice mají bohaté stránky s někdy i tisíci podstránkami/produkty, ale co se týče textu na těchto stránkách, ten bývá často pro vyhledávače žalostný. Internetové obchody totiž častokrát zkopírují popis produktu pouze ze stránek výrobce a jejich stránka je poté považována za duplicitní. Není se pak čemu divit, že po projití 100 až 200 takovýchto stránek crawler většinou odmítá pokračovat e své práci a přesune se na další web.

Jak vypadá správný obsah?

Kvalitní obsah webových stránek se musí rozhodovat mezi kompromisem uspokojení crawlera a návštěvníka. Crawler má všeobecně rád hodně textu, kdežto návštěvníkovi může být často na obtíž dlouhý text, který obsahuje jen část jeho hledaných informací.

Obsah by měl být věcný a měl by korespondovat se zaměřením stránky a klíčovými slovy na ní uvedenými. Takže například prodejce hokejového vybavení může na svých stránkách zveřejnit různé studie kvality jednotlivých částí výzbroje či otevřít diskuze k jednotlivým produktům. Nebo zveřejnění recenzi alias zkušeností s konkrétním produktem napomůže obsahu a zároveň uspokojí návštěvníky.

Obsah také musí být relevantní ke klíčovým slovům. Pokud je na stránkách prodávajících pletivo uvedena jen fotografie pletiva, ale není zde textová informace o pletivech, crawler jen těžko zjistí, že se jedná o stránky s pletivem, i když se návštěvníkům budou zamlouvat.

Jak již bylo řečeno, aktuální obsah je velice důležitý. Crawleři si totiž všímají stáří a změn dokumentů a upřednostňují ty aktuálnější. Aktuální informace navíc mohou být zkombinovány s ostatními technologiemi, jako je například RSS, a být automaticky zasílány na ostatní stránky. Tím lze získat nové návštěvníky a síť zpětných odkazů.

Vaughn Aubuchon (http://www.vaughns-1-pagers.com):

> *„Časté aktualizace - častější indexace - novější cache.“*

Aktuální obsah je také důležitý z hlediska návštěvníků. Nic nedegraduje stránky v očích návštěvníka tak, jako když v sekci novinek vidí čtyři roky staré informace. Takovéto stránky nevzbuzují přílišnou důvěru. Některé stránky to dokonce řeší tím způsobem, že než aby uváděly takto staré novinky, raději neuvádějí žádné.

Obsah by měl také pozitivně propagovat stránky či firmy a vytvořit tak podhoubí pro případnou budoucí spolupráci či prodej. Nemělo by se zapomínat na to, že crawleři se řídí logikou a algoritmy, kdežto lidé emocemi, osobními potřebami a preferencemi. Pokud se podaří nalézt zlatou střední cestu při tvorbě obsahu, tak se jedná o správnou cestu k vytvoření opravdu kvalitního obsahu jak pro vyhledávače, tak pro návštěvníky.

Pohled návštěvníka

Při tvorbě textu by se měl textař vtělit do kůže návštěvníka a zjistit, co na stránce bude hledat za informace. Pokud se jedná o vyhledávač, nesmí zapomenout aby text obsahoval správné množství klíčových slov ve správném tvaru či pádu k indexaci.

Pro lidi je navíc potřeba, aby byl text rozdělen do logických částí, obsahoval hledané informace, byl smysluplný a inspiroval k akci (nákup, spolupráce, registrace, atd.).

Pohled novináře

Při tvorbě textu by se mělo postupovat obdobně, jako postupují novináři při tvorbě článků. Jde o to, že při běžné mluvě se začíná zvolna, až se řečník dopracuje k závěru, ze kterého většinou vyústí tížená informace. U novin je tomu přesně naopak, novinář se snaží hned na začátku zaujmout, aby zaujal čtenáře. Při tvorbě textu pro webové stránky je tomu stejně. Textař by se měl hned soustředit na to podstatné, aby návštěvníkovi pomohl v rozhodování, zda na stránkách zůstat. Jedná se o tzv. systém obrácené pyramidy. [2]

Soubor otázek jako pomoc při psaní textu:

- Kdo?
- Co?
- Kdy?
- Kde?
- Jak?
- Proč?

Pokud se textaři za pomocí klíčových slov a pyramidového systému podaří zodpovědět zmíněný soubor otázek, měl by mít smysluplný text, který bude atraktivní jak pro vyhledávače, tak pro návštěvníky webu.

Odkazy v textu

Odkazy by se neměly omezovat pouze na menu stránky. Vhodné je zařazení odkazů rovnou do textu stránky, čímž se napomůže crawlerům v procházení webu a zařazení klíčových slov obsažených v anchor textu k odkazovaným stránkám. Navíc, crawleři rádi vidí, že se odkazy vyskytují přímo v textu a návštěvníci webu tuto funkčnost ocení především možností okamžitého přesunu na požadovanou stránku. [4]

10 kroků při tvorbě kvalitního textu pro webové stránky

1. Vytvoření osnovy (může kopírovat strukturu webu).
2. Rozdělení textu do jasných a ucelených částí. V podstatě by měly být shodné s body osnovy.
3. Delší texty rozdělit do několika odstavců kdy každý odstavec pojednává o jedné věci.
4. Ještě delší texty nepojednávající o úplně stejné věci rozdělíme na stránky.
5. První věta či souvětí na stránkách by mělo být shrnutí o čem stránka je a další by jej měly rozvinout.
6. Navazující věty psát jako souvětí. Násilná souvětí rozdělit do vícero vět.
7. Ve větě by neměl chybět podmět a přísudek.
8. K zpřehlednění článku bychom měli používat seznamy a odrážky.
9. Doplňovat v textu odkazy na související články a zdroje.
10. Neopakovat stejná slova příliš často.

Na co nezapomenout

- Nepoužívat blízko sebe podobná slova.
- Nebát se hovorových a odborných výrazů.
- Podívejme se na text očima návštěvníka.
- Důležité slova umisťovat tak, aby byla schopna navazovat více blízkých spojení s ostatními slovy.
- Psát ve stejné osobě.
- Závěrem je dobré, přečíst si text nahlas.

Možné negativa

- Příliš dlouhý text.
- Nepochopitelná a zamotaná souvětí.
- Přeskakující myšlenky.
- Příliš se opakující slova.
- Pravopisné chyby.
- Nesprávné interpunkce.

Rychlé informace

Lidé mají rádi rychlé informace, chtějí se co nejdříve dozvědět požadovanou informaci a vyhledávače se snaží napodobovat lidi. Jde tedy o to, snažit se co nejrychleji poskytnou údaje o tom, jaké informace se na stránkách nachází. K tomuto účelu slouží titulek a nadpisy. Lidé často stránky pouze sanují a v 73% případů se na stránkách nezdrží déle než 30 sekund. Proto také vyhledávače přikládají titulkům a nadpisům tak velkou váhu.

Úvodní stránka

Úvodní stránka by měla být užitečná jak pro návštěvníky, tak pro crawlery. Je tedy vhodné vyvarovat se Flashovým či obrázkovým úvodním stránkám, které sice mohou vzbudit dobrý dojem v návštěvníkovi, ale crawler takovou to stránku moc kladně neohodnotí. Faktem je, že právě úvodní stránka je často stránkou, na kterou ostatní weby odkazují.

Úvodní stránka by měla obsahovat stručné informace o firmě a její činnosti s odkazy na ostatní stránky věnující se problematice podrobněji. Nevhodné jsou únavné a zdlouhavé informace typu „historie firmy“ apod.

2.6.4 Black Hat SEO

Tato kapitola se zaměřuje na podvodné taktiky SEO. Není to ovšem návodem, jak tyto techniky praktikovat, jako spíše upozornění pro nezkušené optimalizátory, kteří by si mohli třeba i neúmyslně takto nechat zabanovat jejich stránky. Tyto praktiky nám sice mohou pomoci při SEO, avšak jedná se již o techniky, které jsou za pomyslnou hranicí a pokud je tedy vyhledávač odhalí, zabanuje naši stránky nebo rovnou celou doménu. [1] [2]

Cloaking neboli podstrkávání

Jde většinou o podstrkávání jiného textu vyhledávačům, než vidí návštěvníci. Jde tedy o jasné dosažení lepších výsledků ve vyhledávání, bez nutnosti změny viditelného obsahu. Podstrkávání se většinou provádí serverovým skriptem, robot pak dostává vysoce optimalizovanou stránku.

Další verzí cloakingu je podstrkávání obsahu vyhledávačům, který se návštěvníkům zobrazí až po uhrazení poplatku. Často je takovýto text skrytý před návštěvníky pomocí kaskádových stylů nebo JavaScriptu.

Při nabídnutí neviditelného obsahu vyhledávačům se však nemusí vždy jednat o podvod. Avšak vyhledávače tohle velice nerady vidí, jelikož se snaží poskytovat jejich uživatelům relevantní výsledky, proto dochází k okamžité penalizaci stránek.

Často se podobných technik využívá stránek psaných například ve Flashi či obrázkových stránek. Je potom na uvážení vyhledávače, zda stránky zařadí zpět do svého indexu. Nicméně, i pokud vyhledávač zařadí stránky zpět, jedná se o zdlouhavý proces.

Doorway page neboli podvodné vstupní stránky

Jedná se opět o uměle vytvořenou stránku pro vyhledávače. Jejím cílem je získání lepšího umístění a často obsahuje nezměrné množství klíčových slov, které jsou součástí textu, který nedává smysl. Tyto stránky můžeme také najít pod názvy jako entry pages, bridge pages či gateway pages.

Často se také jedná o tzv. „Made for AdSense stránky“, které obsahují množství odkazů z PPC systému. Cílem takovýchto stránek je potom nalákat uživatele na klíčové slovo, které se vyskytuje v nekvalitním a smysl nedávajícím textu a poté stránka počítá s tím, že uživatel použije některý z odkazů, z čehož má provozovatel takovýchto stránek profit.

Dalším případem doorway stránek je stránka, které v sobě obsahuje iframe, který je viditelný pouze návštěvníkům. Vyhledávačům je potom viditelná pouze vysoce optimalizovaná stránka, která s tou skutečnou, na kterou se dostane návštěvník, nemusí mít nic společného.

Třetím případem je stránka, která po nějaké době návštěvníka automaticky přesměruje na stránky, které nejsou optimalizované.

Dále existují duplicitní stránky, které se snaží dostat do výsledků vyhledávání na různé klíčové slova a poté přesměrují návštěvníka na jednu konkrétní stránku.

Deceptive redirect neboli klamavé přesměrování

Tato technika se často vyskytuje na stránkách s tématikou sexu, gamblerství, warez, apod. Jde o klamavé přesvědčení uživatele, pomocí titulku stránky, že se jedná o to, co hledá a poté jej stránka automaticky přesměruje na nežádoucí stránku. Většinou se jedná o přesměrování JavaScriptem, jelikož kdyby šlo o „klasické“ přesměrování, roboto by obdržel kód 301 v hlavičce a věděl by o tom, že dané stránka přesměrovává na jinou.

Podvodné můžou být i samotné odkazy, kdy odkaz ve skutečnosti neodkazuje na stránku která je zapsaná v atributu `href`, ale na stránku která je zapsaná JavaScriptem u tohoto odkazu.

Hidden content neboli skrytý obsah

I když to může působit nezávadně, je ï tohle vyhledávači považováno za podvodnou techniku. A není se čemu divit, přece jen uživatel opět vidí jiný obsah stránek, než vyhledávač. Mezi nejčastější techniky skrývání textu potom patří:

- Použití stejné barvy textu jako barvy pozadí
- Umístění textu pod obrázek
- Skrytí textu pomocí CSS
- Nastavení nulové velikosti písma

Často k této technice přistupují tvůrci stránek, kteří nejsou schopní do textu stránek zahrnout klíčové slova, ať už z důvodu přání zákazníka či designového omezení stránek. Někdy se jedná teoreticky o zcela nevinnou techniky, kdy například tvůrce stránek napíše název firmy do nadpisu `h1`, aby tak

přidal stránce na důležitosti, následně nadpis skryje a nahradí ho logem firmy, které sice obsahuje stejný text ve formě obrázku, ale tohle už vyhledávač nerozezná.

Podobně je tomu se skrýváním odkazů. Jedná se o odkazy, které má zachytit a zpracovat crawler, ale pro návštěvníka mají zůstat neviditelné či nedostupné. Skrytý odkaz bývá:

- Vytvořený skrytým textem
- Zmenšený pomocí CSS na velikost 1px
- Schovaný v malém znaku (například pomlčka nebo tečka)

Alt tag spamming neboli zaspamování popisu obrázků

Jde o další způsob, který využívají lidé snažící se dostat na jejich stránky co největší počet klíčových slov. Jde tedy konkrétně o zaspamování alternativních textů u obrázku, tedy atributu `alt`. Google například tomuto atributu přikládá docela velkou váhu, mimo jiné je to dáno také díky jeho vyhledávače obrázků, který právě vyhledává v tomto atributu.

Stuffing neboli opakování a matoucí slova

Je to podobné jako zaspamování jako zaspamování popisu obrázků nebo skrytý text, který většinou obsahuje množství smysl nedávajících klíčových slov. V tomto případě se jedná o doplnění textu klíčovými slovy, které následně v kontextu nedávají smysl. U stránek autobazaru by se mohlo jednat například o umělé vkládání fráze „levný autobazar“ do textu.

Stuffing se však nevyskytuje jen u samotného textu stránky, k zaspamování může docházet i v elementech `meta` typu `keywords` a `description` či v titulku stránky `title`. Přičemž zaspamování `keywords` a `description` nám zase tolik nepomůže, ale můžeme tak docílit zabanování stránek. Co se elementu title týče, ten nám sice pomůže, ale můžeme se těšit na okamžitou penalizaci.

Link farms neboli odkazové farmy

Jedná se v podstatě o podobný princip jako u doorway stránek. Jde o několik set nebo i tisíc stránek, které jsou vzájemně provázány odkazy. Vyhledávač tak stránky najde, zaindexuje, a protože obsahují velký počet zpětných odkazů, umístí je ve výsledcích vyhledávání relativně vysoko. Návštěvník je potom při návštěvě takovéto stránky většinou přesměrován na stránku například JavaScriptem nebo jinou metodou. Takovéto stránky se často vyskytují na subdoméně a při jejích penalizaci je tedy umožněn rychlý přesun stránek.

Hrozí zde i nebezpečí pro majitele „poctivých stránek“. Nebezpečí nehrozí, pokud vede odkaz z link farmy na poctivou stránku, ale pokud je tomu naopak, vyhledávač to neuvidí rád. V podstatě existují dva způsoby, jak si poškodit stránky odkazy:

- Text odkazu je skrytý.
- Odkaz odkazuje na stránku používající techniky Black Hat SEO.

Duplicitní stránky

Pokud vyhledávač narazí na dvě různé URL adresy se stejným obsahem, není z toho zrovna nadšený. Při takovémto zjištění sáhne do svého indexu, zjistí, která adresa je důležitější a té druhé si přestane všímat. Vyhodnocení takovéhoto duplicitního obsahu většinou probíhá automaticky na straně vyhledávače a soustředí se na duplicitu obsahu na úrovni:

- URL adres
- Celé subdomény
- Celé domény

Kopírování a vykrádání cizích webů

Obecně vyhledávače velice negativně hodnotí vykrádání cizích textů. Jde doslova o zkopírování textu z jiné stránky a vložení na námi vytvořenou. Tato metoda je rozhodně méně pracná, než vytvoření vlastního textu. K identifikaci takovéto stránky dochází ve dvou případech, buďto si jej vyhledávač sám všimne, nebo jej může upozornit autor originálního textu. V takovém případě se kopírující stránka dostane na černou listinu, ze které se jen těžko dostává zpět do kvalitních pozic ve výsledcích vyhledávání.

Falešné a konkurenční metaznačky

Jedná se o podobnou techniku jako při kopírování textu cizích stránek, s tím rozdílem, že se kopíruje pouze text metaznaček. Jde tedy o další případ parazitismu s klamáním návštěvníka, jelikož obsah metaznaček většinou nesouvisí s obsahem stránek.

Spam klíčovými slovy

Jedná se o podobnou taktiku jako u falešných a konkurenčních metaznaček. Jde o zaspamování metaznačky `keywords` velkým množstvím klíčových slov, z nichž jich je velká část duplicitní a většinou nikterak nesouvisí s obsahem stránek.

Zneužití cizího jména a značek v systémech PPC

Jedná se o nekalou praktiku, kdy firmy zneužívají jména známých značek k propagování vlastní stránky. U nás je to celkem hojný problém, pravděpodobně pro to, že za něj nehrozí postih. Může se tak například stát, že při vyhledávání slova „seznam“ se v Googlu na placených pozicích zobrazí poskytovatelé internetového připojení, kteří fungují pod zcela jinými klíčovými slovy a jiným foremním jménem.

Komentářový a katalogový spam

K této technice není asi moc co dodat. Princip této techniky vyplývá již ze samotného názvu. Často tuto činnost provádí automatizované systémy a programy, které zaspamovávají nejčastěji nechráněné blogy či katalogy.

Postoj vyhledávačů

Jak vlastně vyhledávače zjišťují techniky BH SEO a jak k nim přistupují? Existují dva způsoby, kterými vyhledávač zjistí, že se jedná o BH stránky:

- Automatický
- Ruční

U automatického způsobu většinou parser kontroluje obsah stránek na známé techniky BH SEO a případně stránku penalizuje.

U ručního zjišťování většinou vyhledávač reaguje na upozornění zvenčí, tedy nejčastěji naštvaného majitele podvedených stránek nebo naštvaného uživatele vyhledávače, který se dostane na stránky, které vůbec nepožadoval.

3. Implementace

Tato část diplomové práce se bude zabývat vlastní implementací aplikace. Problémy se kterými jsem se potýkal a jak jsem je vyřešil. Aplikace se skládá ze dvou oddělených částí, z logické a prezentační. Logickou částí je crawler běžící na pozadí serveru, který se stará o stahování, parsování webů a provádění příslušných výpočtů a porovnávání. Presentační částí už je pak jen aplikace, ve které si uživatel zobrazí výsledky. Samozřejmostí je databázový server, který uchovává data.

3.1 Technologie

3.1.1 Logická vrstva

Jako technologii logické vrstvy byla zvolena Java SE7. Důvodem pro zvolení právě Javy je především kvalita více vláknových aplikací v Javě. I když v úvahu pro implementaci crawlera připadal taky C++, vzhledem k rychlosti výpočtů, avšak implementace více vláknových aplikací v C++ je přeci jen náročnější než v Javě. Dalším plus pro Javu ve fázi rozhodování byl fakt, že pro práci s webovými stránkami je dostupných mnohem více open source knihoven pro Javu nežli pro C++. Posledním důležitým faktorem byl fakt, že jsem při vývoji své aplikace spolupracoval na vytvoření rozsáhlejšího projektu s více studenty a po dlouhé debatě jsme se rozhodli právě pro vývoj v Javě, kdy jsme tak usnadnili práci správce serveru, který tak nemusí spravovat každou aplikaci v jiném jazyce, ale vystačí si se správou pouze jednoho Jazyku.

3.1.2 Systém řízení báze dat

Jako systém řízení báze dat (dále jen SŘBD) bylo zvoleny MySQL Server. I když by Oracle mohl být vzhledem k rychlosti výpočtů lepší volbou, tak co se týče licence, je MySQL zdarma, což nás vedlo k rozhodnutí právě pro tento server.

3.1.3 Prezentační vrstva

U výběru prezentační vrstvy sehrál při rozhodování výběru vhodného jazyku největší roli tým, kterého jsem byl součástí. I když u logické vrstvy by nebyl zas tak velký problém, kdyby někdo pracoval v Javě a někdo v C++, tak presentační vrstvu jsme už museli sjednotit do jednotného výstupu. Prvotní podmínkou bylo, aby byl výstup online, dostupný na webu. Jasnou volbou tedy bylo HTML spolu s JavaScriptem. I když se zde nabízí možnost Javy či Silverlightu, přeci jen rozšířenost a podpora HTML je značně větší. Dostali jsme se tedy k otázce výběru jazyku, který poběží na serveru. Hlavními aktéry byl Python a PHP vedle ASP.NETu a Javy, kdy jsme se nakonec rozhodli pro PHP spolu s Nette, kvůli počtu serveru podporujících právě PHP vůči serverům podporujícím ostatní jazyky. Další významnou přidanou hodnotou pro PHP byl fakt, že jsme všichni měli právě s tímto programovacím jazykem největší zkušenosti.

3.2 Implementace logické vrstvy

3.2.1 Stahování stránek

Pro efektivitu je parsování stahování stránek prováděno paralelně. Avšak toto rozhodnuti kromě naproti zrychlení výkonu aplikace nutnost synchronizace, aby stránky nebyly stahovány vícekrát vícero vlákny. Bylo tedy nutné vytvořit zásobník URL adres, který obsahuje adresy, které mají být stažené, avšak neobsahuje adresy, které již byly stažený. Aplikaci se proto uchovává seznam URL adres, které již byly stažené, a při přidávání URL adresy do zásobníku se kontroluje, zda se tato adresa nenachází již v seznamu stažených adres. Později se k tomuto seznamu stažených URL adres přidává samotný obsah stránek na těchto adresách, ale k tomuto se dostanu později. Tento seznam URL adres a zásobník obsahující URL adresy pro stažení, musely byt vytvořeny tak, aby nedocházelo ke kolizím při přístupu vícero vláken k těmto seznamům najednou. Jedná se tedy o synchronizaci, aby se zabránilo výjimce ConcurrentModificationException.

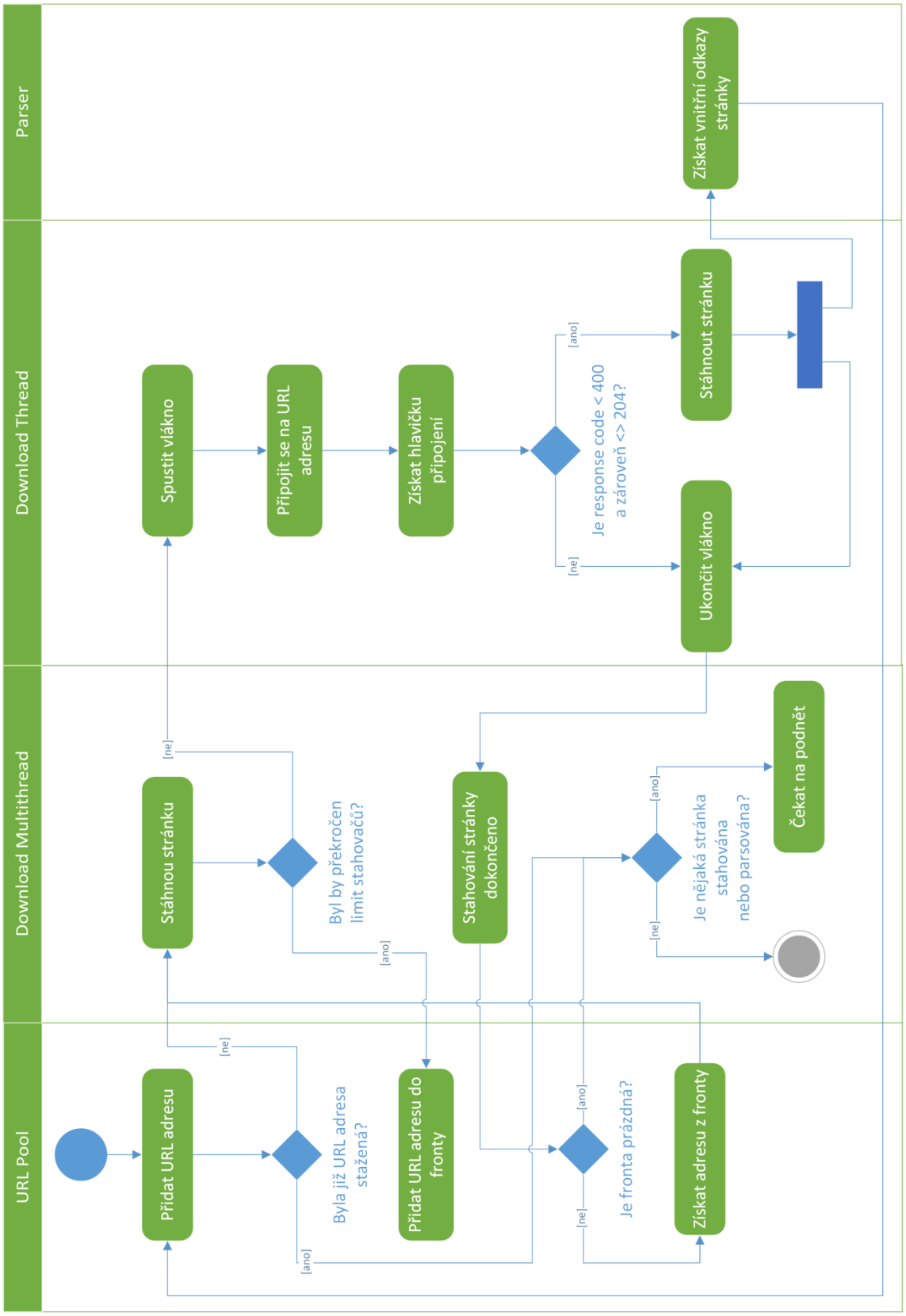
Stahování obsahu webových stránek je prováděno pomocí HttpURLConnection, které je součástí Javy. Pro zvýšení rychlosti stahování jsou stránky komprimovány pomocí ZIPu na straně serveru a následně dekomprimovány při stažení. Je zde ovšem nutnost, aby tuto volbu server, ze kterého je stránka stahována, podporoval. Před stažením stránky se ještě kontroluje odpověď serveru, zda se stránka fyzicky na serveru vůbec nachází.

Stránka je stažená jako pole bytů, buďto pomocí GZIPInputStream nebo čistě jako InputStream, a poté je předána parseru pro získání vnitřních odkazů stránky, které budou následně také staženy, textu a klíčových slov na stránce.

Při stahování více vlákny se objevil problém, kdy servery blokovaly aplikaci z důvodu mnohonásobného stahování stránek. Musel jsem tedy ošetřit, by aktivně stahoval jen určitý počet vláken. Sérií pokusů jsem dospěl k závěru, že optimálním maximálním počtem vláken je pět. Důvod je ten, že pokud jsem zvýšil počet vláken na šest, rychlost aplikace se již dramaticky nezvýšila, ale občas se objevil problém zablokování serverem. Naproti tomu, při omezení počtu vláken na čtyři se výkon oproti pěti vláknům zmenšil, avšak k blokování u pěti vláken již nedocházelo.

Vytvořil jsem tedy třídu spravující vlákna, která při ukončení vlákna stahujícího stránku spustí další vlákno pro stažení další stránky. Hlídá si však, aby počet současně stahujících vláken nebyl větší než pět. Může nastat situace, kdy všechny vlákna stahování ukončí, ale parser může později narazit na stránku, která ještě nebyla stažena. V tomhle případě se třída kontrolující multithreading opět probudí a zpustí nové vlákno, které stáhne data z nově získané URL adresy.

Na diagramu aktivit je zobrazen nástin principu fungování více vláknového stahování stránek. Dovolím si pouze upozornit, že diagram se zaobírá logikou stahování, nikoliv parsování nebo ukládání stránek, upozorňuji především pro to, že by se parser mohl, ač tomu tak není, jevit příliš jednoduše. Parser má samozřejmě za úkol mnohem více činností, nežli jen extrahování vnitřních odkazů Navíc se jedná také o další více vláknovou část aplikace, ale o parseru více v další kapitole.



Obrázek 6: Diagram aktivit – stahování stránek

3.2.2 Parsování stránek

Parser, stejně jako stahovač, neběží jako jedno vlákno, ale jako více vláken. Jakmile vlákno pro stahování stáhne stránku a pošle ji parseru, parser spustí nové vlákno pro parsování získaného textu. Zde již není potřeba omezovat počet vláken, jelikož tato část aplikace již běží v off-line režimu, avšak pro zlepšení výkonnosti aplikace a možnosti koordinace jsem nastavil omezení vláken. Důvod je především ten, že jsem chtěl zamezit tomu, aby současně běželo například tisíc vláken, které by pasovaly stránky. Počet parsujících vláken jsem tedy omezil na 128. Parser má hned několik úkolů které se musí postarat. Jedná se v podstatě mimo jiné o srdce samotné analýzy, jelikož nejenže parsuje stránky, ale také vytváří a analyzuje klíčové slova, kontroluje, zda nejsou na stránce chyby, ať už pravopisné či v HTML kódu, zjišťuje, jaké vidí stránku vyhledávače, atd. Všechny takto získané data jsou uložené do seznamu obsahující stažené URL adresy k příslušné URL adrese. Následně další část aplikace tento seznam prochází a ukládá do databáze, ale tohle je součástí další kapitoly.

Získání textu stránky

Jak bylo již zmíněno, parser dostane stránku od stahovače stránek jako pole bytů. Vyvstává zde hned první problém, jímž je korektní převedení pole bytů na string. Problém je to především pro to, že aby bylo správně převedeno pole bytů na string, musí být definované kódování stránky. Pro získání znakové sady stránky, musí být stránka nejprve převedena z pole bytů na string. Je to tedy jakýsi „začarovaný kruh“. Tento problém jsem vyřešil způsobem, že stránku převádím na string hned dvakrát. Poprvé převede parser pole bytů na string s kódováním UTF-8. Jelikož je pole bytů převáděno po řádcích, upravil jsem tento převod tak, aby se při každém novém řádku podíval, zda tento řádek neobsahuje informaci o kódování stránky. Pokud tuto informaci obsahuje, je buďto zahájeno nové převedení pole bytů na string již se správným kódováním, nebo se pokračuje v kódování, pokud je kódování stránky UTF-8 (což je nejčastějším kódováním HTML stránek). Výhodou tohoto způsobu je, že kódování je obvykle uváděno na začátku HTML stránky (v hlavičce), takže nedochází k procházení celé stránky. Pokud se objeví případ, kdy na stránce není nalezena informace o kódování, je stránka ponechána ve formátu UTF-8 i za předpokladu, že může být ve skutečnosti kódování jiné.

Informace od vyhledávačů

Prvním, co parser provede po získání textu (mohl by to provádět i před získáním), je zjištění informací o dané stránce od vyhledávačů. Oněmi informacemi je míněn S-rank od Seznamu a Alexa Rank. Původně aplikace zjišťovala i PageRank od Googlu a datum poslední návštěvy stránky některým z crawlerů od Googlu, avšak Google aplikaci banoval pro velké množství dotazů. Musel jsem se tedy uchýlit k řešení, kdy se ptám jen na PageRank jedná stránky, většinou úvodní, a stejně tak i na datum poslední indexace úvodní stránky.

Porovnání s HTML kódem ostatních stránek

Aby nedocházelo ke zbytečnému ukládání duplicitních textů stránek do seznamu stažených stránek (tedy případ, kdy jsou dvě či více stránek identické, ať už kompletně či jen textem, který parser zajímá), je porovnáván nejprve md5 kód HTML stránek (32 místný jedinečný kód generovaný pro HTML kód stránky) s kódy ostatních stránek v seznamu.

Pokud je nalezena shoda s některou ze stránek v seznamu, je uložena informace k parsované stránce, respektive URL adrese o tom, se kterou stránkou je identická a práce tohoto konkrétního vlákna parseru je u konce. Pokud není nalezena shoda, parser pokračuje dále.

W3C validace

Dalším krokem ve zpracování stránky je ověření, zda je HTML kód validní. K tomuto účelu byla použita již existující open source knihovna pro Javu zvaná ReXSL, konkrétně její modul W3C. Největším problémem, se kterým jsem se u tohoto řešení potýkal, bylo sehnání všech závislých knihoven, které pro svou funkčnost ReXSL potřebuje. Některé totiž již nejsou oficiálně podporovány a jejich stažení bylo tak komplikovanější. Nicméně, podařilo se mi sehnat všechny potřebné knihovny a ReXSL zprovoznit.

Parsování

Nyní dochází k samotnému parsování stránky. K tomuto účelu jsem využil existující knihovny Jsoup, které slouží k parsování HTML stránek a navíc dokáže převádět relativní URL adresy na absolutní. Ještě nežli se Jsoup pustí do samotného parsování, je potřeba z textu stránky odstranit komentáře. Po odstranění komentářů již aplikace přechází k parsování.

Jako první se ze stránky parsují **vnitřní odkazy**, které jsou uloženy do zásobníku adres čekajících na stažení, pokud již nebyly stažené. Důvod, proč se odkazy parsují jako první je zjevný a to, aby se zajistil co nejrychlejší chod aplikace, aby stahovač zbytečně nečekal na rozparsování celé stránky pro pokračování v práci.

Po získání vnitřních odkazů dochází na parsování celé stránky. Parser neparsuje každý element, ale jen ty elementy, které jsou významné pro vyhledávače, neboli významné pro SEO. Jednotlivé elementy nebudu vypisovat, dostanu se k nim při vypisování, které informace se ukládají do databáze. Nicméně, parsuje jak hlavičkové elementy, tak elementy v těle stránky.

Další věcí, kterou parser na stránce vyhledává je skript, podle kterého pozná, zda je stránka kontrolována systémem Google Analytics, či nikoliv.

Posledním úkolem samotného parsování je zjištění, zda se na stránce nenacházejí zastaralé elementy anebo atributy. Jedná se například o elementy jako `font`, či atributy jako `align` apod. Nedá se říci, že pokud se na stránce tyto tagy vyskytují, že tj. vyloženě špatně (nemůžeme tedy stránku označit jako nevalidní), ale je dobré na tuto skutečnost upozornit, jelikož vyhledávače tyto tagy nerady vidí.

Parser také u všech textových elementů a atributů provádí kontrolu pravopisu, a pokud narazí na chybu, uloží si informaci, na které pozici se dané slovo s chybou nachází. Kontrola pravopisu je prováděna pomocí knihovny Jazzy. Jazzy nebylo rozhodně první volbou, ke které jsem se uchýlil. Důvodem je, že se jedná o knihovnu, která je sice poměrně zastaralá a primárně podporuje jen angličtinu, avšak je zde relativně snadné importovat slovník, díky kterému bude podporovat češtinu. Následně stačí udržovat aktuální pouze tento slovník a aplikace bude fungovat správně. Potřeboval jsem tedy pouze sehnat slovník českých slov. K získání tohoto slovníku jsem využil aplikace Aspell,

do které jsem stáhl český slovník a pomocí příkazu v příkazové řádce vygeneroval slovník, který je kompatibilní s Jazzy.

Další možnosti, o které jsem se při kontrole pravopisu pokoušel, byly následující:

- Aspell – Jedná se sice o kvalitní knihovnu s podporou češtiny, zároveň relativně aktuální, avšak je určena pro C++. I když jsem vytvořil rozhraní mezi Javou a C++, nebylo toto řešení ideální.
- Pspell – Je součástí Aspellu, tedy opět určeno pro C++, byl zde tedy stejný problém.
- Hunspell – Opět stejný problém jako u předchozích knihoven a to, že nepodporuje češtinu.
- Dále uvedu seznam knihoven, které jsou psané pro Javu, avšak neobsahují češtinu. Jedná se tedy o: LanguageTool, JSpell, JaSpell, Jortho.

Porovnání textu s ostatními stránkami

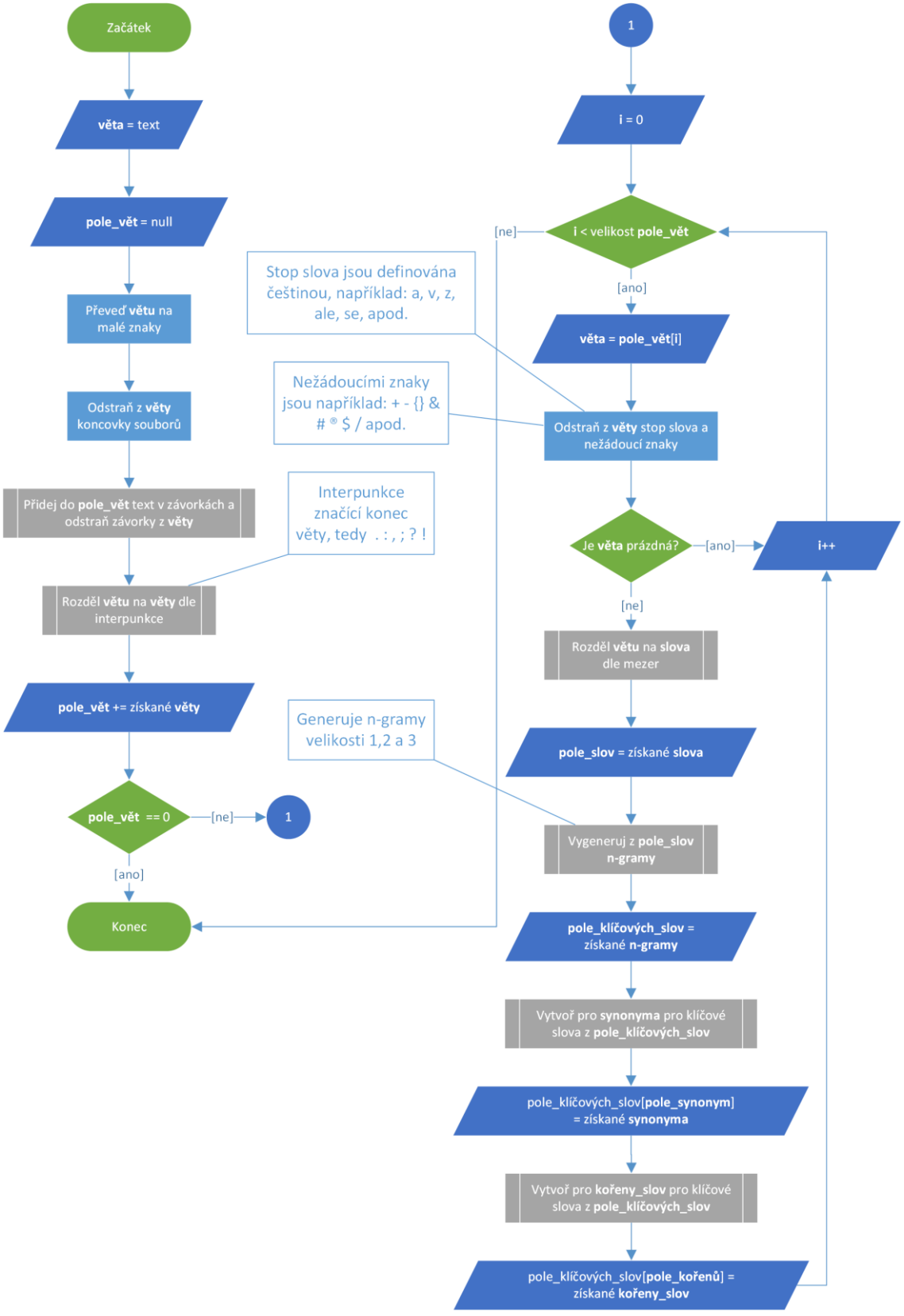
Nyní dochází k podobnému kroku jako u porovnávání HTML kódů. Vyparsovaný, a tedy významný text, je seřazen podle tagů ze kterých byl získán, seřazen podle abecedy, vložen do jednoho stringu a zakódován do md5 kódu. Takto získaný kód se porovnává s ostatními, již staženými stránkami, čímž se docílí porovnání obsahu stránek. Zabrání se tak tomu, abych se zbytečně vytvářely klíčové slova pro stránky s duplicitním obsahem. Jeli tedy obsah duplicitní, vlákno zaznamená informaci o duplicitě k dané URL adrese a ukončí svou činnost.

Generování n-gramů neboli klíčových slov

Následuje poslední část, kterou parser provádí, jíž je generování n-gramů neboli klíčových slov. Generovány jsou jen n-gramy velikosti 1,2 a 3. Důvodem, proč tomu tak je, že ani samotné vyhledávače negenerují delší n-gramy.

N-gramy jsou generovány z textů získaných po parsování stránky, tedy jen z elementů významných pro SEO analýzu. Samotné generování není však až tak jednoduché, nemůžeme vzít text a vygenerovat z něj slova. Existují zde určité pravidla, kterými se řídí vyhledávače a kterými se tedy musí řídit i tato aplikace.

- Převedení textu na **malé znaky**.
- Odstranění **stop slov** a **koncovek souborů** z textu. Stop slova jsou slova, jako například spojky či předložky. Tedy slova, neudávající konkrétní význam výslednému klíčovému slovu.
- Vytažení textu ze **závorek**. Jedná se o interpunkci, která nám říká, že text v závorkách je odlišný od textu mimo něj, nemůžeme tedy slučovat slova v závorkách se slovy mimo závorky. Proto musíme z textu odstranit závorky a udělat z obsahu těchto závorek vlastní text.
- Rozdělení textu podle jazyk specifikované **interpunkce**, jako například tečky, čárky, středníky. Důvod je jasný, nechceme generovat klíčové slovo, které bude obsahovat poslední slovo z jedné věty a první slovo z následující věty.



Obrázek 7: Vývojový diagram – generování klíčových slov

Po ošetření textu, tedy získání vět pro generování klíčových slov, aplikace přechází k samotnému generování. Negeneruje však jen jedno klíčové slovo, ale k tomuto klíčovému slovu si vygeneruje i všechny kombinace jeho synonym, aby při ukládání do databáze mohla být provedena analýza nejen nad samotným klíčovým slovem ale i nad synonymy tohoto klíčového slova. Jako další se klíčové slovo upraví na klíčové slovo, které se skládá jen z kořenu slova. Tedy slovo, ze kterého jsou odebrány přípony a předpony. Takže například ze slova „lesník“ dostaneme slovo „les“.

Pro generování klíčových slov jsem vytvořil vlastní třídu, která pracuje ze synonymy. Slovník českých synonym jsem vyextrahoval z knihovny českého tezauru. Jedná se o 70 souborů obsahujících slovo a k němu existující synonyma. Při spuštění aplikace dochází k nahrání těchto souborů do paměti plus zaznamenání prvních slov vyskytujících se v jednotlivých souborech. Je zde tedy 70 počátečních slov, podle kterých lze určit, ve kterém slovníku hledat požadované slovo. Tímto se značně urychlí vyhledávání, jelikož vyhledáme binárně, který slovník použít a následně v konkrétním slovníku binárně vyhledáme požadované slovo. Binární vyhledávání je umožněno, jelikož slovníky i slova v těchto slovnících jsou abecedně seřazena. Pokud není slovo nalezeno, nemá synonyma.

Získané klíčové slovo, jeho synonyma a klíčové slovo složené z kořenu slov jsou uloženy do seznamu k příslušné URL adrese s údajem, ze kterého elementu na stránce bylo získáno.

Dokončení parsování

Nyní je parsování a sním spjaté úkony u konce. Vlákno předá informaci o ukončení třídě spravující vlákna parseru, aby mohla spustit další vlákno pro parsování a parsovat tak další stránku. Dále předá informaci třídě, která je zodpovědná za ukládání stránek do databáze.

3.2.3 Uložení dat do databáze

Po dokončení parsování přichází na řadu ukládání dat do databáze. Nejedná se však jen o čisté ukládání, ale v některých případech i malá úprava dat, jelikož není vše ukládáno do paměti, aby se šetřilo místem.

Prvním způsobem ukládání, které jsem odzkoušel, bylo uložení všech dat až po stažení a rozparsování všech stránek. Způsob to je docela elegantní a rychlý, jelikož lze využít „multi insertů“, což značně urychlí ukládání dat. Tento způsob fungoval bezchybně, avšak při práci s většími weby se objevil problém nedostatku paměti, neboli OutOfMemoryError. Problém je v tom, že na serveru jsem omezen maximální pamětí 1GB a data z rozsáhlých webů tuto kapacitu jednoduše přetekla.

Musel jsem tedy přistoupit k řešení, kdy po stažení stránky okamžitě ukládám data do databáze a uvolním místo v paměti. Aby tato metoda byla maximálně efektivní, musel jsem vytvořit dávkovače insertů (pro rychlejší vkládání dat nežli po jednom) a zajistit synchronizaci mezi vlákny, aby k sdílené tabulce mezi vícero vlákny přistupovalo vždy jen jedno vlákno.

Samotné data získané ze stránky se ukládají do dvou databází. První databází je databáze slov, obsahující pouze vygenerované n-gramy, která je sdílená mezi všemi projekty (alias weby). Tato databáze je pojmenována jako „seodic“. Kdy *seo* je prefix pro všechny mnou vytvořené databáze a

tabulky (jelikož jak jsem se již zmiňoval, pracujeme v teamu a moje aplikace není jediná, která pracuje s databází) a *dic* je zkratka odvozená z anglického slovíčka „dictionary“, tedy slovník.

Druhou databází, do které je stránka ukládána, je databáze samotného projektu, alias webu, či domény. (Každá doména má svou vlastní databázi.) Jedná se o databázi „seo_číslo“, kdy číslo značí ID které je přiřazeno dané doméně. ID domény jsou uloženy v databázi „seocor“ (*cor* je zkratka anglického core, neboli jádro). Tato databáze obsahuje seznam všech stránek, textu na stránkách, klíčových slov na stránkách, informace o souborech robots.txt, sitemap.xml a informace získané od vyhledávačů. E-R modely databází jsou uvedeny v příloze.

Ukládání stránek

Jak již bylo zmíněno výše, jedná se o ukládání dat do databáze seo_číslo. V této části ukládání toho není moc, co objasňovat. Jednoduše se vezmou data stránky, která jsou uložena v sezamu v paměti. Jedná se tedy o uložení následujících dat:

- Unikátní ID stránky
- URL adresa stránky
- SEO skóre stránky
    - Jedná se o generování skóre, které je v rozmezí od 0 do 100. Počítáme s tím, že každá stránka má skóre 100 a následně ji za každý „přestupek“ vůči SEO odečítáme body. K tomuto výpočtu dochází až po uložení všech stránek, protože mimo jiné je zde zahrnuto například porovnávání hlaviček webových stránek.
- Informaci zda je HTML kód či text stránky identický s jinou stránkou a kterou
- Informaci, zda je stránka přístupná při zohledňování pravidel robots.txt
- Odpověď stránky neboli response code, kódování neboli charset stránky
- Doba stažení stránky
- Velikost HTML kódu stránky
- Md5 kód HTML stránky a md5 kód textu stránky
- Informaci, ze které stránky se parser na danou stránku dostal
- Informaci, zda je stránka monitorována aplikací Google Analytics
- Počet W3C chyb a varování na stránce
- Informaci zda stránka obsahuje zastaralé HTML elementy anebo atributy
- S-Rank a Alexa Rank stránky

Dále jsou ukládány jednotlivé texty získané při parsování stránky spolu s pozicemi případných pravopisných chyb v daných textech:

- Hlavička stránky
    - `Doctype` (bez kontroly pravopisu)
    - Text z `meta` elementu `description`
    - Text z `meta` elementu `keywords`
    - Autor stránky v `meta` elementu `author` (bez kontroly pravopisu)
    - Obsah `meta` elementu `robots` (bez kontroly pravopisu)

    - Jazyk stránky z `meta` elementu `content language` (bez kontroly pravopisu)
    - Titulek stránky z elementu `title`
    - Nadpis stránky z elementu `h1` – i když se nejedná o HTML element obsažený v hlavičce stránky, tak je na stránce jedinečný, proto je zařazen do tabulky obsahující hlavičkové informace
- Tělo stránky
    - Nadpisy `h2` – `h1`
    - Odstavce `p`
    - Zvýraznění textu `strong`, `b`, `em` a `i`
    - Seznamy `li`, `dd` a `dt`
    - Odkazy na stránce, tedy element `a`
        - Anchor text odkazu
        - Obsah atributu `title`
        - Obsah atributu `href`, kdy je zvlášť ukládána doména a samotná cesta (bez kontroly pravopisu)
    - Obrázky na stránce, element `img`
        - Obsah atributu `alt`
        - Obsah atributu `title`
        - Obsah atributu `src`, opět je zvlášť ukládána doména a cesta (bez kontroly pravopisu)

Jako poslední informace k příslušné stránce jsou ukládány samotné n-gramy, tedy ID těchto n-gramů, které jsou uložené v databázi *seodic*. Jedná se tedy o uložení následujících informací ke každému n-gramu:

- ID n-gramu
- Velikost n-gramu (1,2 nebo 3)
- Element, nebo atribut, ve kterém se nachází – tato hodnota se mění s uložením každého n-gramu. N-gram se tedy může například v odstavci nacházet na stránce 10 krát.

Optimalizace a problémy při ukládání informací o stránce

Pro rychlejší ukládání dat ze stránky, je pro každou tabulku vytvořen zásobník insertů, který obsahuje informace o budoucím generovaném dotazu. Jakmile je obsah celé stránky zpracován a převeden do zásobníku jsou provedeny hromadné inserty.

Kapacita zásobníku je omezená, jelikož u některých stránek docházelo opět k přetečení paměti. Omezení je nastaveno na 1000 údajů pro každou tabulku. Ve většině případů stránka neobsahuje více než 1000 elementů jednoho typu (například odkazů), avšak vyskytly se takovéto případy a proto je zajištěno ošetření.

Ošetření ukládání obsahu elementů a tagů je značně jednoduší nežli ukládání n-gramů na stránce, jelikož u n-gramů na stránce se jednotlivé již uložené data mohou měnit (zvyšovat, či snižovat počet výskytu daného slova v daném elementu na stránce). Je zde tedy nutnost, kromě přidávání do seznamu

také vyhledávání v tomto seznamu a případná modifikace. Aby se dosáhlo co největšího výkonu, jsou tedy data v tomto zásobníku seřazena podle ID n-gramu, což nám zajistí možnost binárního vyhledávání v tomto seznamu. Zde jsem narazil na problém u Javy, jímž je, že datové kolekce, které dědí z kolekce Collection nepodporují binární vyhledávání, ale dovolují, aby se k prvkům přistupovalo pomocí ID v seznamu. Naproti tomu kolekce dědící z kolekce TreeSet podporují binární vyhledávání, avšak jako výsledek nevrací nalezený prvek, nýbrž jen informaci, zda se v seznamu nachází, či nikoliv.

Musel jsem tedy vytvořit vlastní kolekci, která kombinuje výhody obou těchto kolekcí. Jedná se tedy o kolekci, ze které můžeme vybírat prvky kolekce pomocí pozice v seznamku a zároveň v ní můžeme binárně vyhledávat. Binární vyhledávání nám sice nevrací nalezený prvek, ale vrací nám pozici prvku v seznamu, což nám bohatě vystačí. Na druhou stranu, musí být zajištěno, aby byl seznam uspořádaný (pro binární vyhledávání), což využívá určitý výpočetní výkon. Tato kolekce však není využívána jen v tomto případě ukládání, ale také v případě ukládání slov do databáze slovníků. Nicméně, i když musí být zajištěno seřazení seznamu, je v závěru aplikace s binárním vyhledáváním v některých případech i o polovinu rychlejší než s iteračním vyhledáváním. Tento fakt hodně závisí na počtu slov na stránce, čím větší je počet, tím větší je rozdíl mezi binárním a iteračním vyhledáváním. Což není vzhledem k složitostí binárního vyhledávání *log n* vůči lineárnímu vyhledávání *n* překvapivé.

Ukládání n-gramů

Jak jsem se již zmínil výše, n-gramy se ukládají do speciální databáze slovníku obsahující slovník slova a slovníky kombinací těchto slov. Celkem se tedy jedná o 13 slovníků.

- Slovník obsahující slova délky 1, který jim přiřazuje ID, pro efektivnější práci s n-gramy.
- 6 slovníky obsahující abecedně neseřazené slova n-gramů
    - Rozlišující X nerozlišující diakritiku
    - Zohledňující synonyma a respektující diakritiku X zohledňující synonyma a nerespektující diakritiku
    - Pouze kořeny slov respektující diakritiku X pouze kořeny slov nerespektující diakritiku
- 6 slovníku, kdy jsou slova n-gramů seřazeny podle abecedy
    - Těchto šest typů slovníku je totožných s šesti typy slovníků, které mají neseřazené slova n-gramů podle abecedy

K synonymům je třeba podotknout, že nejsou ukládány všechny kombinace všech nalezených synonym. Ukládání probíhá tak, že je hledáno jakékoliv synonymum k danému v tabulce synonym. Pokud je některé ze synonym nalezeno, změní se slovo za nalezené synonymum. Pokud není synonymum v tabulce, je uložen základní tvar slova do tabulky synonym. Tím se zredukuje množství slov ukládaných do databáze. Pokud je například do tabulky synonym uloženo slovo „auto“ a později bude ukládáno slovo „automobil“, tak se slovo „automobil“ neuloží, ale zamění se za slovo „auto“. Aby bylo později možno dohledat, které synonymum patří ke kterému slovu, je do tabulky s původním slovem uloženo ID synonyma ke každému slovu. Toto synonymum může být identické se slovem.

Při ukládání n-gramů stránky je zde opět zásobník s binárním vyhledáváním, které je popsáno výše, avšak je zde jeden rozdíl. Ukládání dat do slovníku je sdíleno mezi všemi stránkami, jedná se tedy o objekt sdílený mezi všemi stránkami. To přináší nový problém, čímž je synchronizace tohoto objektu pro přístup vícero vláken. Avšak nárůst výkonu je značný, jelikož nemusíme provádět vložení dat do databáze při dokončení ukládání každé stránky, ale pouze při dosáhnutí limitu zásobníku, nebo při uložení celého webu. Na druhou stranu, více záznamů v paměti znamená pomalejší vyhledávání, ale zase je tím značně ulehčena práce databázového serveru, který by jinak musel neustále odpovídat na existenci daného slova v databázi.

Pro zvýšení efektivity jsou tedy všechny slova ze slovníku obsahujícího slovo a jeho ID seřazena podle abecedy a uložena do paměti. Tento seznam je abecedně seřazen podle textu slov. K seřazení dochází již na straně databázového serveru, což se ukázalo jako značně rychlejší řešení nežli seřazovat seznam za běhu programu až po získání všech slov. Poté musí aplikace pouze zajistit, aby byl seznam seřazený i po přidání nového slova.

Ostatní tabulky neboli jednotlivé slovníky, pak obsahují pouze ID konkrétního slova a na které pozici v n-gramu se nachází. Každý takovýto n-gram je v každém slovníku jedinečný a je mu přiřazeno unikátní ID. V případě těchto slovníku neukládáme data z databáze do paměti, jelikož tyto slovníku jsou relativně rozsáhlé a je jich dvanáct. Při pokusu o uložení více než 40 000 záznamů z každého takovéto slovníku v paměti docházelo k přetečení paměti. Zásobníky těchto slovníku jsou tedy nastaveny na maximální kapacitu 40 000 záznamů, což u menších webů bohatě stačí, avšak u větších dochází k průběžnému ukládání a uvolňování paměti. V těchto zásobících je jsou n-gramy také vyhledávány binárním vyhledáváním a je to zajištěno tím, že n-gramy jsou seřazeny podle ID slov a jejich pozice. Jednoslovné n-gramy jsou tedy výše nežli víceslovné n-gramy.

Zároveň je třeba podotknout, že do těchto zásobníků jsou ukládány pouze záznamy, které se již nenacházejí v databázi. Dochází tedy k větší zatíženosti databázového serveru neustálými dotazy o existenci n-gramu v daném slovníku. Avšak pro alespoň nějaké urychlení se aplikace první podívá do paměti, zda se tam již n-gram nenachází. Počítá se zde s tím, že se n-gramy budou na jednotlivých stránkách webu opakovat. Pokud se však web ukládá již podruhé, většinou se v samotné paměti moc n-gramů nenachází, protože jsou již uloženy v databázi. Pokoušel jsem se urychlit vyhledávání tím, že pokud jsem n-gram nalezl v databázi, uložil jsem jej do paměti s informací, že se nemá znovu ukládat do databáze. Dosáhl jsem tím menšího zatížení databázového serveru. Avšak aby tato metoda byla efektivní, potřeboval bych větší příděl paměti a tím i možnost větších zásobníku. Jde o to, že pokud se n-gram nevyskytuje na dané stránce vícekrát, zaberu výpočetní výkon tím, že jej musím seřadit v seznamu v paměti, ale za chvíli se tento seznam zase vyprázdní při ukládání do databáze a poté jej opět naplňuji stejnými slovy. Závěr z tohoto řešení tedy je, že u menších webů je tato metoda efektivnější, ale u rozsáhlejších webových stránek není efektivní. Jelikož se počítá s tím, že aplikace bude procházet primárně internetové obchody, které jsou opravdu rozsáhlé, upustil jsem od tohoto řešení a vrátil se zpět k dotazování serveru a ukládání pouze nových záznamů do paměti.

Uložení informací o doméně

Jakmile jsou uloženy všechny stránky a n-gramy přichází na řadu uložení informací o doméně. Jedná se o informace typu:

- Soubor robots.txt
- Soubor sitemap.txt
- Informace o serveru – IP a lokace

Jelikož není vytvářena nová databáze *seo_číslo* pro každý nový projekt, ale pouze pro každou doménu, musí být zajištěno, aby nebyly ukládány redundantní data (neboli stránky, které byly již uloženy v minulosti). Proto má každá doména zároveň různé verze, které obsahují ID jednotlivých informací:

- Aktuálních pravidel v souboru robots.txt
- Aktuálních stránek v souboru sitemap.xml
- Aktuálních přístupných stránek na webu
- PageRank a poslední indexaci Googlem úvodní stránky pro danou verzi
- Datum a čas verze

Commit

V poslední části přichází na řadu commit, tedy potvrzení uložení dat do databáze. Pokud se tedy během ukládání objeví chyba, je proveden rollback a nenastane tak případ, kdy by byly v databázi „mrtvá“ data.

Po provedení příkazu commit je do databáze *seocor* uložena informace o tom, kterou verzi má pro dané nastavení a konkrétní datum stahování webu přiřadit. Vygeneruje si tak vlastní ID projektu, které je jedinečné pro každé stahování (pokud se liší od předchozího). Toto ID je následně předáváno aplikaci, která zajišťuje porovnávání stránek, ta si z tohoto ID zjistí, které stránky má porovnávat.

3.2.4 Porovnání stránek

Aplikace pro porovnání stránek obdrží ID dvou projektů, které má porovnávat a váhy jednotlivých HTML elementů neboli jejich významnost ohodnocenou od 0 do 10. Ze zadaného ID projektu si aplikace vybere jednotlivé stránky obou projektů a provádí porovnávání kombinací všech stránek pro všechny slova všech slovníku na těchto stránkách obsažených.

Pro každé slovo je tedy vybrán výskyt v jednotlivých elementech na obou porovnávaných stránkách. Tento výskyt je pře násoben významností daného elementu, tzn., pokud je n-gram pouze v titulku stránky, ale titulek má významnost 10, je to, jako by byl na stránce deset krát. Takto získané výskyty jsou porovnány a uloženy do databáze do příslušného sloupce v tabulce pro daný slovník. Tabulky obsahují sloupce, které zastupují jednotlivé možnosti každého n-gramu, neboli:

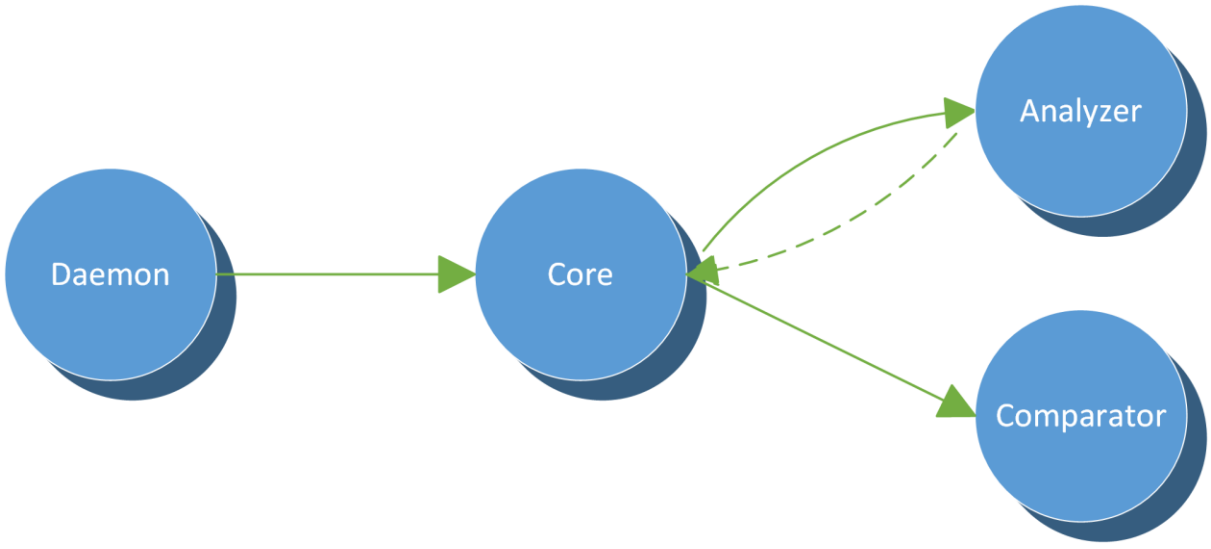
- N-gram se nachází pouze na stránce A nebo na stránce B.
- Výskyt slova se nachází na obou stránkách, ale na stránce A/B je ve větším množství. V tomhle případě musí být uložena informace o tom, v jakém množství se vyskytuje na stránce s menším výskytem.
- Výskyt slova na stránce A a B je stejný.

Pro urychlení porovnávání, jelikož rozsáhlost je poměrně značná, nejsou do databáze ukládány slova, jejichž výskyt po převážení je u obou stránek menší než 9. Tím se šetří místo pro slova, která jsou pro uživatele nevýznamná.

Co se týče zásobníku, zde již nedochází k ukládání vícero dat najednou. Důvodem je to, že pokud aplikace používá zásobníky jako v předchozích případech, ukládání je pomalejší, nežli ukládání po jednom slově. Důvodem je pravděpodobně to, že stejně jako u stahování, parsování a ukládání stránek i zde je porovnávání prováděno více vláknově, což zajišťuje, že během provádění výpočtů porovnávání může vlákno, které již porovnávání dokončil uložit informace do databáze a databáze tak zbytečně „nespí“ a nečeká na příval velkého dotazu.

3.2.5 Jednotlivé aplikace logické vrstvy

Logická vrstva se dohromady skládá ze čtyř aplikací. První dvě aplikace jsem již popsal výše, první aplikací tedy je aplikace stahující, parsující a ukládající informace o stránce. Druhou aplikací je aplikace, která porovnává jednotlivé weby, respektive projekty neboli verze webů.



Obrázek 8: Aplikace logické vrstvy

Třetí aplikací je aplikace, která spouští první aplikaci pro každý web, který obdrží pro zpracování a po zpracování všech těchto webů spouští druhou aplikaci pro kombinace referenčních webů a konkurenčních webů, které mají být porovnávány.

Poslední, čtvrtou aplikací, je daemon, který běží na pozadí serveru a čeká na nové požadavky. Jakmile obdrží požadavek pro zpracování projektu, zjistí, jaké webové stránky tento projekt obsahuje, do jaké hloubky mají být jednotlivé stránky procházeny a jaké jsou nastaveny významnosti jednotlivých elementů. Po zjištění těchto informací spustí třetí aplikaci a tyto údaji jí předá. Následně pouze čeká na dokončení stahování a porovnávání a pak se vrací do stavu očekávání na další projekt.

3.3 Implementace prezentační vrstvy

Implementace presentační vrstvy nebyla ani zdaleka tak složitá, jako implementace logické vrstvy. Důvodem je to, že v presentační vrstvě se provádí jen minimální počet výpočetních úkonů, vesměs se jedná pouze o získání dat z databáze a jejich správné naformátování.

Kromě výstupu obsahuje prezentační vrstva také modul pro spuštění projektu, je tedy nutné, aby zde bylo nastavení významnosti jednotlivých elementů. Významnost je defaultně předvyplněná hodnotami, které jsem odvodil při studování problematiky SEO, avšak uživatel si může tyto hodnoty změnit.

V presentační vrstvě se také nachází modul pro administrátora aplikace, který může spravovat stop slova, koncovky či nežádoucí znaky, které budou vyloučeny z textu při tvorbě n-gramů.

Pro zpřehlednění dále popisovaných výstupů z aplikace uvedu **mapu webu**, tedy mapu výstupu. Na výstupu máme v podstatě dva hlavní moduly:

- SEO
    - Povolené stránky
        - Detailní informace o stránce a o textu na stránce
        - Výskyt klíčových slov na stránce
        - Copywriting
    - Zakázané stránky
    - Stránky v Sitemap.xml
    - Pravidla v souboru robots.txt
- Podobnost
    - Porovnávané stránky pro jednotlivé velikosti n-gramů
        - Detail porovnání a možnost modifikace výpočtů

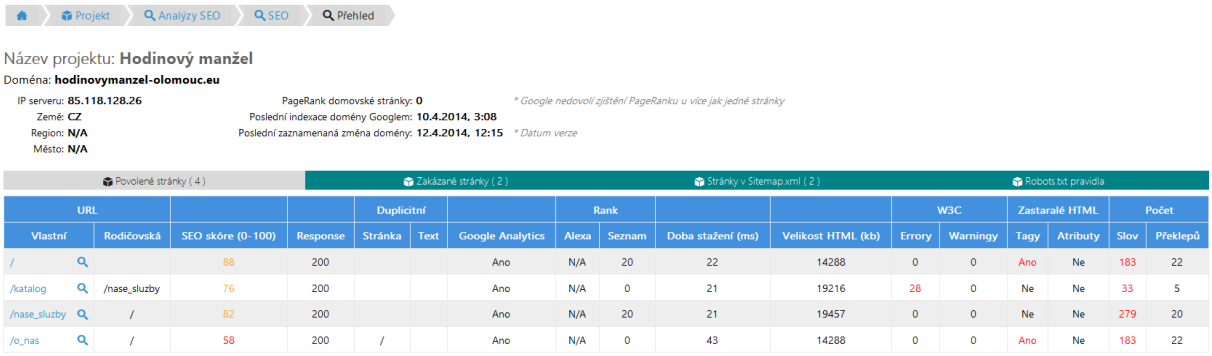
3.3.1 SEO

Na úvodní stránce se nachází seznam všech verzí projektů, tedy časově odlišných testů. U každé verze je na prvním místě tabulka referenčních webu, za kterou následuje tabulka konkurenčních webů. U každého webu je udána URL adresa, počet stránek na daném webu a celkové SEO skóre, které je vypočítáno zprůměrováním skóre všech stránek na webu. Po kliknutí na skóre daného webu je uživatel přesměrován na další stránku, stránku se všemi „povolenými“ stránkami na webu.

Povolené stránky

Zde se nachází seznam všech stránek na webu. Jsou zde základní informace jako URL adresa stránky, respektive její cesta. Dále pak, ze které stránky se robot na danou stránku dostal a SEO skóre. Mimo tyto základní informace jsou zde uvedeny detailnější informace o stránce, jimiž jsou:

- Odpověď stránky neboli response code.
- Zda je HTML kód stránky nebo pouze text stránky s některou stránkou na webu identický a pokud ano, tak informace o tom, o kterou stránku se jedná.
- Informace, zda je stránka monitorována systémem Google Analytics
- S-Rank a Alexa Rank stránky
- Doba stažení a velikost HTML kódu stránky
- Počet W3C chyb a varování na stránce
- Zda se na stránce vyskytují zastaralé HTML elementy nebo atributy
- A nakonec počet slov na stránce plus počet překlepů



URL				Duplicitní			Rank				W3C		Zastaralé HTML		Počet	
Vlastní	Rodičovská	SEO skóre (0-100)	Response	Stránka	Text	Google Analytics	Alexa	Seznam	Doba stažení (ms)	Velikost HTML (kb)	Errory	Warningy	Tagy	Atributy	Slov	Překlepů
/		88	200			Ano	N/A	20	22	14288	0	0	Ano	Ne	183	22
/katalog	/nase_sluzby	76	200			Ano	N/A	0	21	19216	28	0	Ne	Ne	33	5
/nase_sluzby	/	82	200			Ano	N/A	20	21	19457	0	0	Ne	Ne	279	20
/o_nas	/	58	200	/		Ano	N/A	0	43	14288	0	0	Ano	Ne	183	22

Obrázek 9: Náhled aplikace – SEO, Povolené stránky domény

Po kliknutí na URL adresu (cestu) stránky se je uživatel přesměrován na detailní výpis informací o stránce (bude popsáno později). Na stejné úrovni jako je seznam povolených stránek je také seznam zakázaných stránek, pravidla souboru robots.txt a stránky obsažené v souboru sitemap.xml.

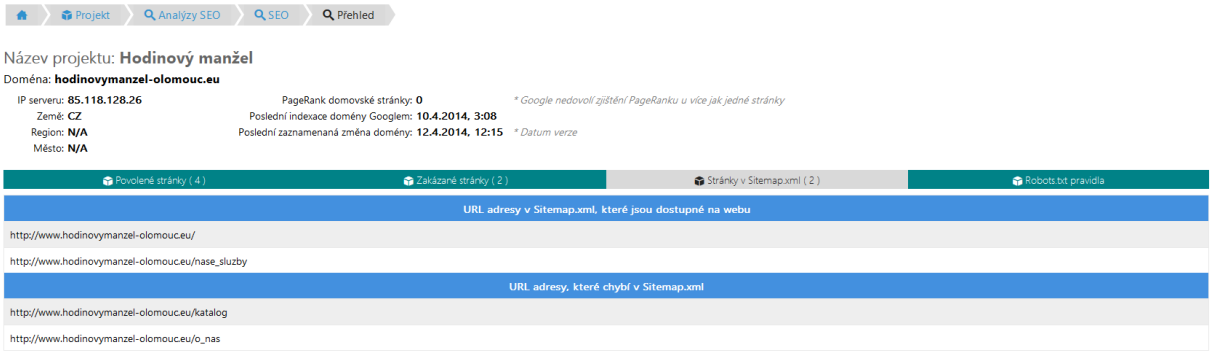
Další informací, která se těchto čtyřech stránkách nachází, je informace o doméně. Tedy IP adresa a lokace domény, PageRank úvodní stránky domény a kdy naposledy Google indexoval úvodní stránku domény. Pokud se na webu nenachází soubor robots.txt anebo Sitemap.xml, je uživatel na tuto skutečnost upozorněn.

Zakázané stránky

Jedná se o stránky, které jsou pro crawlery, tedy i pro tuto aplikaci, v souboru robots.txt zakázány. Aplikace tyto stránky neparsuje, ale pokud narazí na takovouto stránku, uloží si do databáze informaci o URL adrese, která je zakázaná. Uživatel tedy zde nalezne pouze URL adresu, respektive cestu stránky a stránku, na které se crawler o této stránce poprvé dozvěděl.

Stránky v Sitemap.xml

Zde nalezneme hned tři seznamy. Prvním seznamem jsou v podstatě korektní URL adresy, jedná se o URL adresy, které jsou dostupné na webu a zároveň jsou zapsány v souboru Sitemap.xml. Dalším seznamem jsou URL adresy, které jsou dostupné na webu, ale nejsou zapsány v souboru Sitemap.xml. Jedná se tedy o stránky, které v souboru Sitemap.xml chybí. Posledním seznamem jsou naopak URL adresy, které jsou zapsány v souboru Sitemap.xml, ale nejsou dostupné na webu, tedy stránky, které v souboru Sitemap.xml nadbývají.



Obrázek 10: Náhled aplikace – SEO, Analýza sitemap.xml

Pravidla v souboru robots.txt

Zde jsou vypsány pravidla, které byly vyčteny ze souboru robots.txt. Je zde tedy sezam pravidel, zda se jedná o pravidlo disallow (zakázat) či allow (povolit), o jaké pravidlo se jedná, tedy která stránka nebo stránky mají být zakázány a nakonec pro koho je dané pravidlo určeno.

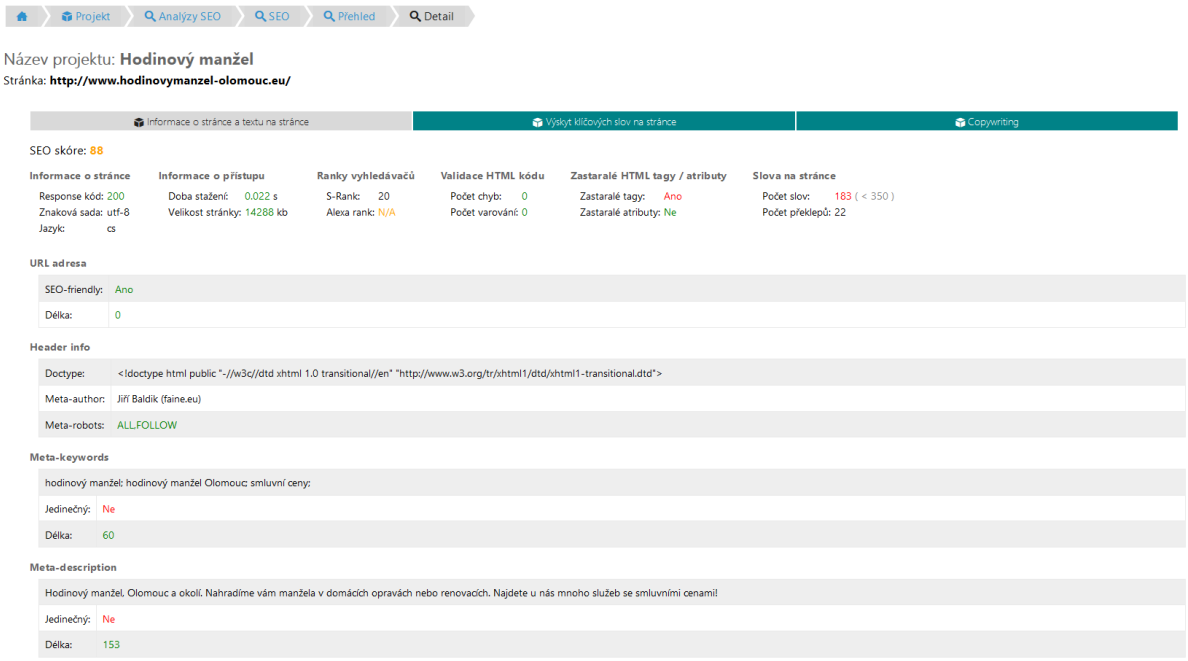
Detailní informace o stránce a textu na stránce

V úvodu neboli hlavičce detailního výpisu jsou zobrazeny informace, které jsou totožné s informacemi z povolených stránek. Tyto informace jsou navíc doplněny informací o znakové sadě stránky a jazyku stránky.

Dále je zde detailní výpisy a informace o jednotlivých elementech a atributech stránky. Jedná se o:

- URL adresu stránky – obsahuje informaci, zda je URL adresa SEO friendly a délku URL adresy. Uživatel je upozorněn na skutečnost, jeli adresa příliš dlouhá.
- Hlavičkové informace – zde se nachází doctype stránky, autor stránky (jeli uveden) a `meta robots` na stránce.
- Meta keywords – jedná se o výpis klíčových slov uvedených v elementu `meta keywords`. U těchto klíčových slov jsou zároveň uvedeny pravopisné chyby. Je zde také uvedená informace, zda jsou klíčové slova jedinečné vůči všem ostatním stránkám domény a zda nejsou příliš dlouhá.
- Meta description – `meta description` má podobný výpis jako `meta keywords` s tím rozdílem, že maximální délka u `meta description` je jiná než u `meta keywords`.

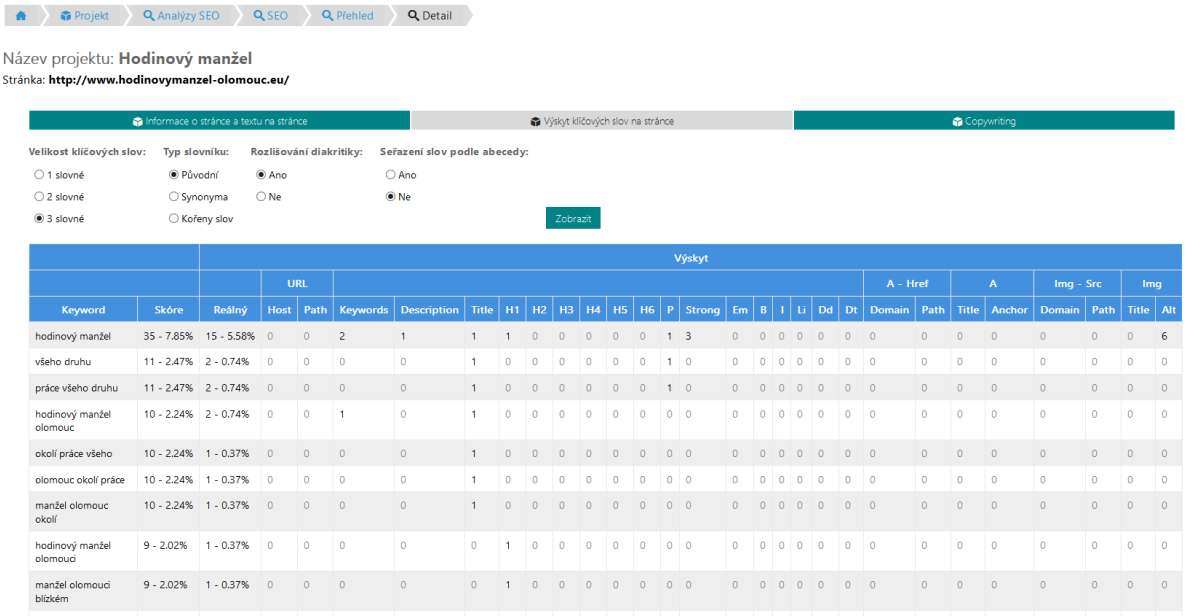
- Title – titulek stránky obsahuje opět totožné informace jako `meta keywords`. Opět se liší maximální délka.
- H1 – informace o nadpisu jsou na tom stejně jako informace o titulku. Samozřejmě se zde opět liší maximální délka.
- Následují výpisy, kde se už nekontroluje délka ani jedinečnost, ale uživatel je upozorněn, má-li v textu pravopisnou chybu. Jedná se o výpisy z elementů `h2` – `h6`, `p`, `strong`, `b`, `em`, `i`, `li`, `dd` a `dt`.
- Posledními výpisy jsou odkazy a obrázky.
    - Odkazy jsou vypisovány zvlášť vnitřní a vnější. Jsou zde uvedeny atributy `href`, `rel`, `title` a samotný text odkazu. U atributu `title` a textu odkazu je kontrolován pravopis. Uživatel je upozorněn, není-li atribut `title` u odkazu definovaný.
    - Stejně jako odkazy i u obrázků je uvedeno, zda je odkaz uvedený v `src` vnější nebo vnitřní. Dále se zde nachází atributy `alt` a `title`, u kterých je kontrolován pravopis a uživatel je upozorněn, není-li některý z atributů definován.



Obrázek 11: Náhled aplikace – SEO, Detailní informace o stránce

Výskyt klíčových slov na stránce

Zde je výpis všech klíčových slov nacházejících se na stránce. Ke každému klíčovému slovu je uveden reálný výskyt, elementy a atributy ve kterých se vyskytuje a množství v jakém se v nich vyskytuje a nakonec skóre každého klíčového slova, které je počítáno jako výskyt v jednotlivém tagu krát významnost daného tagu.



		Výskyt																											
			URL																		A - Href		A		Img - Src		Img	
Keyword	Skóre	Reálný	Host	Path	Keywords	Description	Title	H1	H2	H3	H4	H5	H6	P	Strong	Em	B	I	Li	Dd	Dt	Domain	Path	Title	Anchor	Domain	Path	Title	Alt
hodinový manžel	35 - 7.85%	15 - 5.58%	0	0	2	1	1	1	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	6
všeho druhu	11 - 2.47%	2 - 0.74%	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
práce všeho druhu	11 - 2.47%	2 - 0.74%	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hodinový manžel olomouc	10 - 2.24%	2 - 0.74%	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
okolí práce všeho	10 - 2.24%	1 - 0.37%	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
olomouc okolí práce	10 - 2.24%	1 - 0.37%	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
manžel olomouc okolí	10 - 2.24%	1 - 0.37%	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
hodinový manžel olomouci	9 - 2.02%	1 - 0.37%	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
manžel olomouci blízkém	9 - 2.02%	1 - 0.37%	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Obrázek 12: Náhled aplikace – SEO, Klíčové slova na stránce

Je zde možnost, navolit si, které klíčové slova uživatele zajímají. Základním rozlišením je velikost klíčových slov, tedy klíčové slova délky 1,2 nebo 3. Poté následuje výběr slovníku. Jedná se o výše popisovaných 12 slovníků, přičemž uživatel si pomocí tří kombinací nastavení vybere slovník, který ho zajímá (jedná se o kombinace 3 x 2 x 2).

První kombinací je typ slovníku. Jedná se v podstatě o typ slov v klíčových slovech. Možnosti tedy jsou: původní nezměněná slova, synonyma slov, kořeny slov.

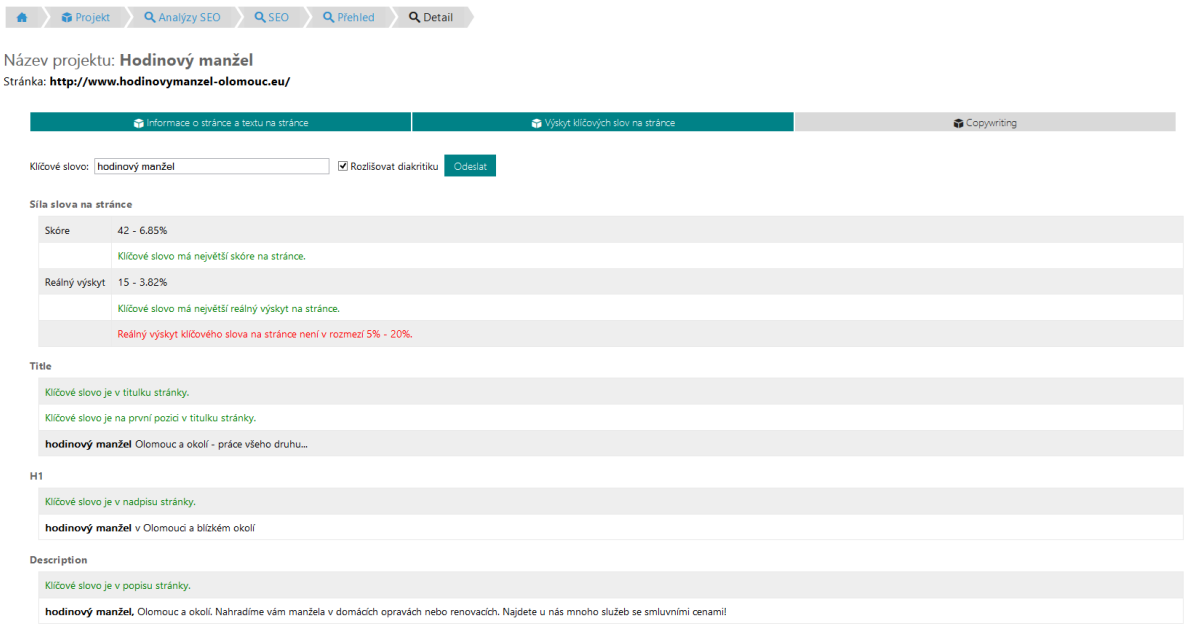
Dalšími dvěma možnostmi je, zda chce uživatel u slov rozlišovat diakritiku, či nikoliv a mají-li být slova obsažené v klíčovém slově seřazena nebo ne. Je jasné, že pokud se jedná o klíčové slova délky jedna, je zde tato možnost zbytečná.

Copywriting

Prvním krokem, který musí uživatel před analýzou copywritingu podniknout je definování klíčového slova a zda rozlišovat diakritiku. Jakmile tak učiní, jsou vygenerovány doporučení, jak změnit text na stránce, aby byla optimalizována na zadané klíčové slovo.

- Síla slova na stránce – zde je uživatel upozorněn, pokud je skóre anebo reálný výskyt klíčového slova na stránce menší než nějakého jiného slova. Také se zde kontroluje, zda je reálný výskyt klíčového slova v rozmezí 5 – 20%.
- Title – pro zpřehlednění je zobrazen text titulku stránky, ve kterém je zvýrazněno klíčové slovo. Pokud se klíčové slovo v titulku nenachází, nebo se nachází, ale není na prvním místě, je uživatel upozorněn.
- H1 – zobrazené informace jsou stejné jako v případě titulku s tím rozdílem, že se nekontroluje, zda je klíčové slovo na prvním místě v nadpisu stránky.
- Informace o description a keywords jsou totožné s nadpisem.

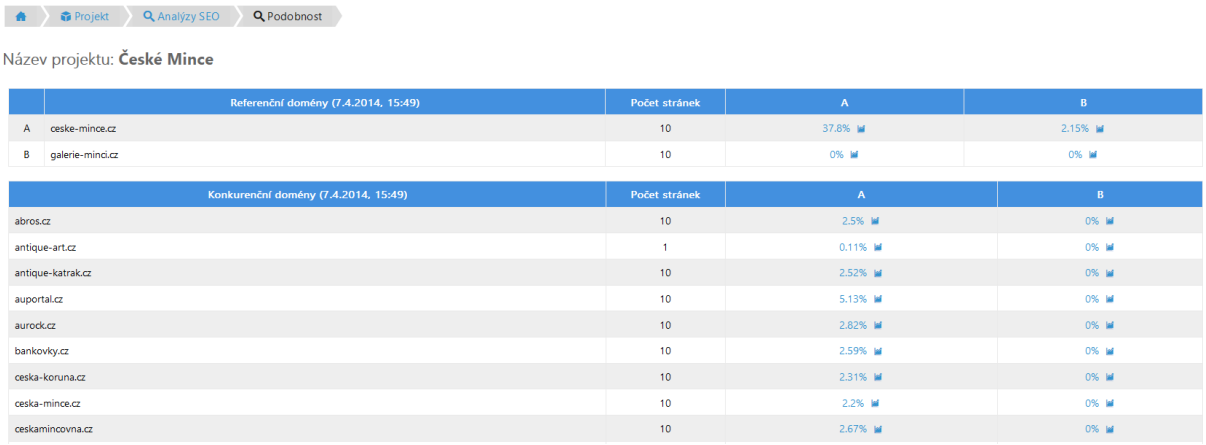
- Strong, Em, B, I – zde je kontrolováno, zda se klíčové slovo nachází v některém ze zvýrazňujících elementů. Zároveň je uživatel upozorněn, pokud se klíčové slovo nachází v elementu `b` místo elementu `strong`. Opět jsou vypsány elementy, které obsahují hledané klíčové slovo. Toto klíčové slovo je v nich pro zpřehlednění zvýrazněno.
- Ostatní tagy – jedná se pouze o výpis obsahu elementů a atributů na stránce, které obsahují klíčové slovo (klíčové slovo je zvýrazněno). Jedná se o elementy `h2` – `h6`, `p`, `li`, `dd`, `dt` a o atributy `img-alt`, `img-title` a `a-title`.



Obrázek 13: Náhled aplikace – SEO, Copywriting stránky

3.3.2 Podobnost

Stejně jako v případě zobrazení SEO jsou také na úvodní stránce při zobrazení podobnosti vypsány jednotlivé verze analýz, tedy verze v čase. Opět jsou zde dva seznamy, první seznam obsahující referenční URL adresy a druhý seznam obsahující konkurenční URL adresy. U všech URL adres je zobrazen výsledek porovnání, tedy podobnost domén, s jednotlivými referenčními doménami. Referenční domény jsou také porovnávány mezi sebou a také jedna referenční doména vůči sobě samé. Po kliknutí na číslo určující podobnost se dostaneme na detailní výpis stránek a jejich podobností.



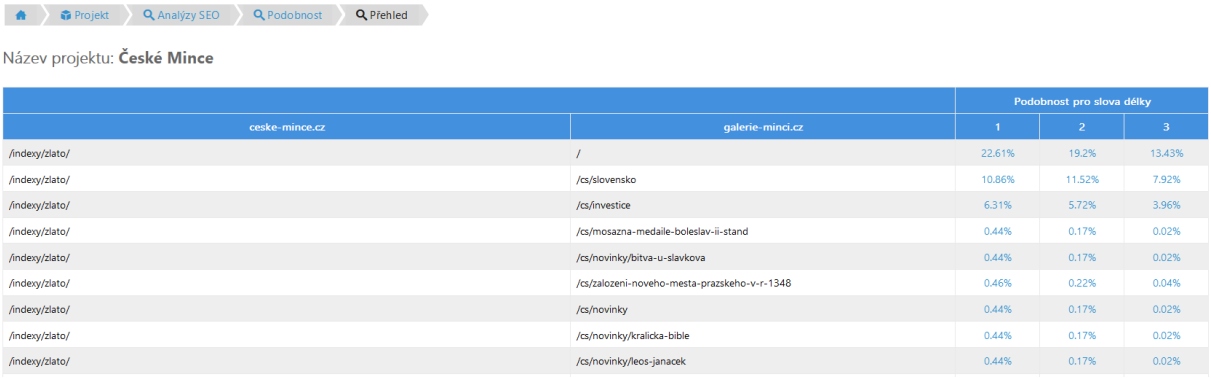
	Referenční domény (7.4.2014, 15:49)	Počet stránek	A	B
A	ceske-mince.cz	10	37.8%	2.15%
B	galerie-minci.cz	10	0%	0%

Konkurenční domény (7.4.2014, 15:49)	Počet stránek	A	B
abros.cz	10	2.5%	0%
antique-art.cz	1	0.11%	0%
antique-katrak.cz	10	2.52%	0%
auportal.cz	10	5.13%	0%
aurock.cz	10	2.82%	0%
bankovky.cz	10	2.59%	0%
ceska-koruna.cz	10	2.31%	0%
ceska-mince.cz	10	2.2%	0%
ceskamincovna.cz	10	2.67%	0%

Obrázek 14: Náhled aplikace – Podobnost referenčních domén vůči ostatním

Podobnost porovnávaných stránek pro jednotlivé velikosti n-gramů

Zde je zobrazen výpis všech porovnávaných dvojic stránek, přičemž je každá stránka porovnávaná s každou. Pokud jde tedy o dva weby, které mají po deseti stránkách, bude zde výsledek 100 porovnání. Ke každé z dvojic stránek jsou uvedeny tři výsledky porovnávání. Jde o výsledky pro klíčové slova velikosti 1,2 a 3. Po kliknutí na tento výsledek porovnávání je uživatel přesměrován na detailní výpis porovnání dvou stránek. Zde si může velikost klíčových slov později změnit.

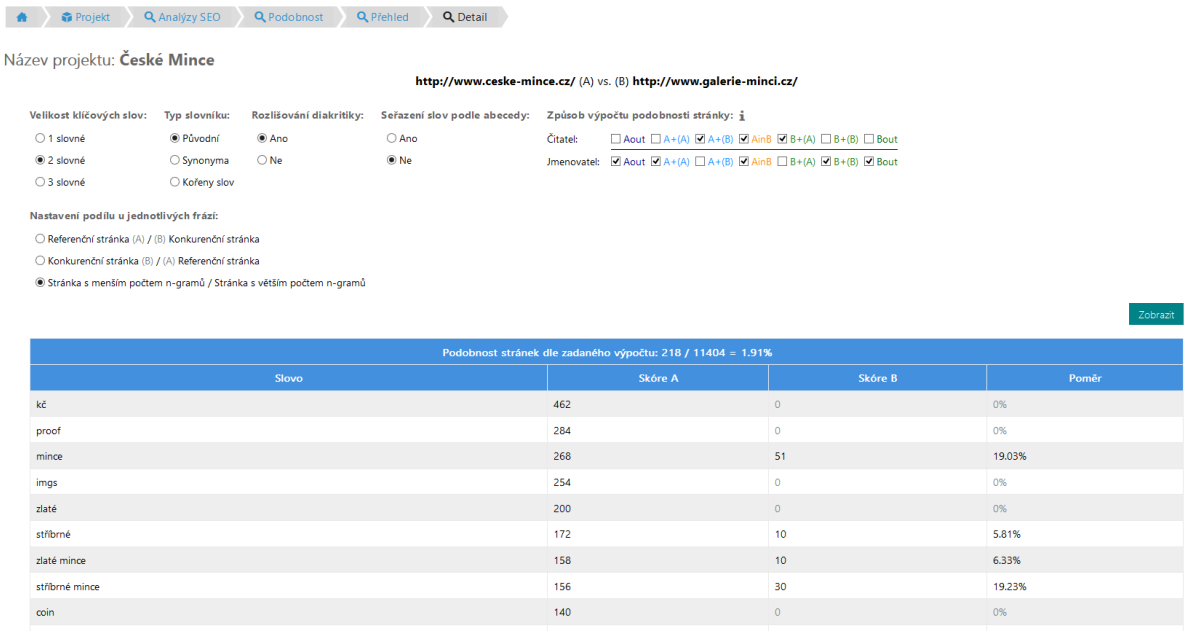


		Podobnost pro slova délky		
ceske-mince.cz	galerie-minci.cz	1	2	3
/indexy/zlato/	/	22.61%	19.2%	13.43%
/indexy/zlato/	/cs/slovensko	10.86%	11.52%	7.92%
/indexy/zlato/	/cs/investice	6.31%	5.72%	3.96%
/indexy/zlato/	/cs/mosazna-medaile-boleslav-ii-stand	0.44%	0.17%	0.02%
/indexy/zlato/	/cs/novinky/bitva-u-slavkova	0.44%	0.17%	0.02%
/indexy/zlato/	/cs/zalozeni-noveho-mesta-prazskeho-v-r-1348	0.46%	0.22%	0.04%
/indexy/zlato/	/cs/novinky	0.44%	0.17%	0.02%
/indexy/zlato/	/cs/novinky/kralicka-bible	0.44%	0.17%	0.02%
/indexy/zlato/	/cs/novinky/leos-janacek	0.44%	0.17%	0.02%

Obrázek 15: Náhled aplikace – Podobnost, Podobnost všech stránek dvou domén

Detail porovnání a možnost modifikace výpočtů

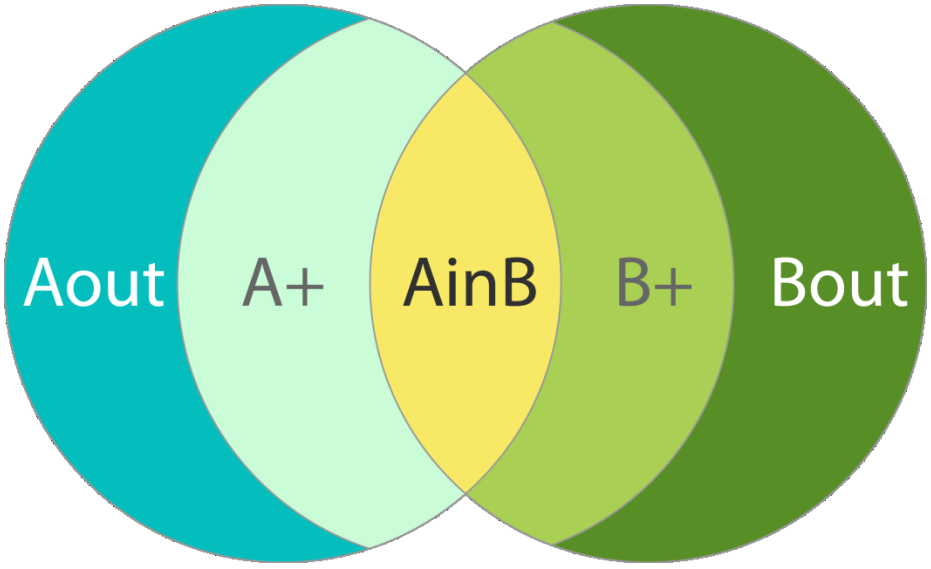
Zde se nachází nastavení možnosti výpočtu podobnosti stránek a jednotlivých klíčových slov. První možnost je známá už z předchozí kapitoly o výpisu SEO, jedná se o volbu velikosti klíčového slova a slovníku. Nebudu tedy již znovu popisovat tuto problematiku.



Obrázek 16: Náhled aplikace – Podobnost, Podobnost dvou stránek a modifikace výpočtů

Výpočet podobnosti stránek

Další možností je způsob výpočtu podobnosti stránky. Výpočet je založen na myšlence Vennových diagramů, avšak pro naše potřeby byla zde přidána ještě jedna možnost. Jedná se tedy o množiny klíčových slov, které se nacházejí na obou stránkách ve stejném množství, nebo na obou stránkách ale v rozdílném množství a jako poslední je, že se nachází jen na jedné stránce.



Obrázek 17: Množina klíčových slov, na dvou doménách/stránkách

Aout	klíčové slova webu A, které se nenachází na webu B
A+	klíčové slova webu A, nacházející se na webu A ve větším množství, než na webu B
AinB	klíčové slova vyskytující se na webu A i B ve stejném, nenulovém množství
B+	klíčové slova webu B, nacházející se na webu B ve větším množství, než na webu A
Bout	klíčové slova webu B, které se nenachází na webu A

Tabulka 4: Legenda k množině klíčových slov, na dvou doménách/stránkách

Všechny tyto množiny, může uživatel přiřadit jak do čitatele, tak do jmenovatele zlomku, který vypočítává podobnost dvou stránek. Zlomek vypočítávající například kolik klíčových slov, mají stránky společných z celkového počtu slov, bude vypadat následovně:

$$\frac{(A+)+(AinB)+(B+)}{(Aout)+(A+)+(AinB)+(B+)+(Bout)}$$

V aplikaci je navíc přidána možnost, kdy uživatel může rozlišit, zda chce počítat s menším nebo větším počtem v případě A+ nebo B+. Tedy, pokud jde například o A+, zda chce zahrnout počet slov na stránce A (větší) nebo na stránce B (menší).

Výpočet podílu klíčového slova na dvou stránkách

Poslední možností je způsob výpočtu podílu klíčových slov. Zde je situace jednodušší. Uživatel má na výběr ze tří možností pro výpočet podobnosti, a to:

- $\frac{Referenční\ web}{Konkurenční\ web}$
- $\frac{Konkurenční\ web}{Referenční\ web}$
- $\frac{Větší\ výskyt\ slova}{Menší\ výskyt\ slova}$

Nejsnadnější je vysvětlení na příkladu:

- Jako referenční web je zvolen web A a jako konkurenční web B
- Reálné klíčové slova, reprezentují znaky m, n, o, p, q

Jako první je definována tabulka, která reprezentuje slova s výskyty. U každého klíčového slova lze určit, do které množiny dané klíčové slovo spadá.

Výskyt na webu A	Klíčové slovo	Výskyt na webu B	Závěr
0	**m**	2	Bout
1	**n**	4	B+
2	**o**	2	AinB
5	**p**	3	A+
3	**q**	0	Aout

Tabulka 5: Příklad - Výskyt frází na stránkách A a B

Nyní je na řadě porovnání výsledků třech možných výpočtů podílu klíčového slova na dvou stránkách. Z předchozí tabulky je znám výskyty jednotlivých klíčových slov, který je zapsán buďto do čitatele nebo do jmenovatele.

Klíčové slovo	Referenční/Konkurenční	Konkurenční/Referenční	Větší/Menší
m	-	$\frac{2}{0}$	$\frac{2}{0}$
n	$\frac{1}{4}$	$\frac{4}{1}$	$\frac{4}{1}$
o	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$
p	$\frac{5}{3}$	$\frac{3}{5}$	$\frac{5}{3}$
q	$\frac{3}{0}$	-	$\frac{3}{0}$

Tabulka 6: Příklad - Zvolený přepočet u výskytu frází na A a B

Nyní nic nebrání tomu, aby byly zvolené přepočty převedeny na procenta, z čehož lze snadněji vyčíst poměr klíčového slova stránky A vůči stránce B.

Klíčové slovo	Referenční/Konkurenční	Konkurenční/Referenční	Větší/Menší
m	-	+	+
n	25%	400%	400%
o	100%	100%	100%
p	166,66%	60%	166,66%
q	+	-	+

Tabulka 7: Příklad: Výsledek porovnání frází na stránkách A a B

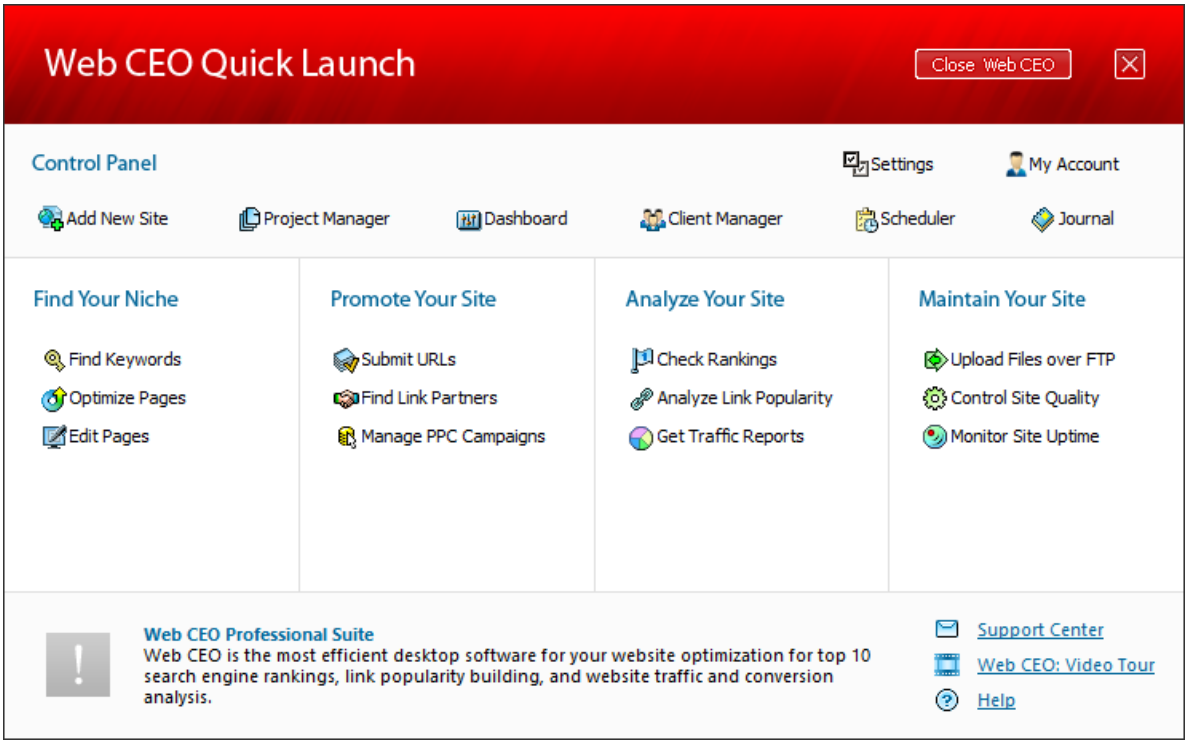
4. Porovnání s existujícími aplikacemi

V této kapitole se zaměřím na porovnání mnou vytvořené aplikace s již existujícími aplikacemi na trhu. Půjde o porovnání přidaných hodnot jednotlivých aplikací, pokud nějaké jsou. Bohužel, většina komplexnějších SEO aplikací není zdarma, nebo jsou zpřístupněny v demo verzi, které zdaleka neobsáhnou všechny funkce, které aplikace v placené verzi nabízejí. Tuto skutečnost je tedy dobré dbát na zřetel. Na druhou stranu, můžeme považovat to, že je aplikace zdarma, jako přidanou hodnotu.

Při hodnocení budou udávány jako zápory funkce, které porovnávána aplikace nepodporuje, ale moje a jako kladné budou udávány funkce, které má porovnávána aplikace navíc nebo kvalitnější oproti mé aplikaci.

4.1 Web CEO

Jedná se o velice komplexní nástroj s velkým množstvím funkcí, které jsou velkým přínosem při SEO. Nabízí se zde funkce od počátečního výběru klíčového slova, přes analýzu vlastních stránek, jak si na tom stojí s klíčovými slovy a kvalitou stránek k analýze vyhledávačů, tedy na jakých pozicích se nachází námi zadané stránky a kontrole zpětných odkazů v internetu. Co se týče porovnání s mou aplikací, mohu se zaměřit pouze na porovnání on-page SEO optimalizace, jelikož analýza off-page faktorů a vyhledávačů nebyla cílem této práce. Musím také zmínit, že tento program nenabízí licenci zdarma, i když nabízí třiceti denní zkušební verzi, která však nemá všechny funkce jako verze placená.



Obrázek 18: Náhled aplikace Web CEO

4.1.1 Control Site Quality

V této části program se nachází hodnocení našich stránek, především co se týče kvality stránek. Tedy zda něco na stránkách nechybí nebo naopak nenadbývá.

Check Site Quality

- Kontroluje velikost stránky, `title`, `keywords` a `description`
- – Ignoruje přesměrování přes .htaccess (hlásí chybějící stránky)
- – Neudává dobu stažení stránky, znakovou sadu, jazyk, response code, W3C chyby a varování, zda stránka obsahuje zastaralé HTML tagy, počet slov a počet překlepů
- – Neporovnává hlavičkové elementy s ostatními stránkami domény pro kontrolu duplicity
- \+ Kontroluje CSS soubory
- \+ Kontroluje, zda element `img` obsahuje tagy `width` a `height`

View Reports

- Počet stránek na webu
- Velikost stránky
- `title`, `keywords`, `description` plus kontrola jejich délky
- Kontrola atributu `alt` u elementu `img`
- – Neuvádí, které stránky jsou povolené a které blokované
- – Neuvádí, které stránky jsou nebo nejsou v sitemap.xml
- – Nevypisuje vnitřní a vnější odkazy
- – Nekontroluje duplicitu `title`, `keywords` a `description`
- – Nekontroluje atribut `title` u elementu `img`
- – Nekontroluje tagy `title`, `rel` a text odkazu u elementu `a`
- – Nezobrazuje obsahy HTML tagů a překlepy v těchto textech
- – Nekontroluje `meta author`, `meta robots` a `doctype`
- \+ Nefunkční odkazy
- \+ Příliš zanořené stránky
- \+ Kontroluje, zda element `img` obsahuje tagy `width` a `height`
- \+ Počet HTML stránek, obrázků, videí a audio souborů na webu

4.1.2 Ranking

Zde se jedná převážně o analýzu vyhledávačů, což nespadá do on-page optimalizace, avšak vypíchnu jednu záložku, jíž je „Indexed pages“.

Indexed pages

- \+ Zobrazuje indexovaná stránky Googlem (O stejnou funkci jsem se pokoušel i u své aplikace, ale bohužel, Google mě banoval za příliš mnoho requestů. Kontroluji tedy alespoň, zda je indexována úvodní stránka.)

4.1.3 Optimization

Část aplikace, která je zaměřená na rady, jak zlepšit optimalizaci stránek na zadané klíčové slovo. Navíc také obsahuje stručný přehled některých údajů z analýz v předchozích dvou kapitolách.

Optimization Advice and Analysis

- Rady jak zlepšit optimalizaci na dané klíčové slovo
- Kontrola klíčového slova v `head` elementech
- Kontrola klíčového slova v URL adrese
- Síla klíčového slova na stránce

– U `body` elementů nevypisuje, kde konkrétně se klíčové slovo nachází nebo ne, tedy kromě `h1` a atributu `alt` u `img`

– Používá pouze jeden slovník, nejde tedy například vybrat, zda chci rozlišovat diakritiku a pořadí slov anebo pracovat se synonymy či kořeny slov

\+ Kontrola zda je na stránce JavaScript

\+ Kontrola, zda nemá pozadí stejnou barvu jako text

\+ Kontrola, zda není použit příliš malý font (viz. Black Hat SEO)

Density Analysis

- Výpis nejsilnějších klíčových slov na stránce

– Výpis všech klíčových slov na stránce

– Volba slovníku, ze kterého čerpat

– Zobrazení, ve kterém konkrétním elementu nebo atributu se klíčové slovo nachází

\+ Klíčové slova délky 4

SE View Report

- Zvýraznění klíčového slova v textu stránky

– Označení překlepů na stránce

– Chybí informace, ze kterého tagu daný text pochází

4.1.4 Závěrečné zhodnocení

Jasnou nevýhodou Web CEO je fakt, že není zdarma. Další velkou nevýhodou pro českého uživatele je fakt, že aplikace není primárně tvořena pro stránky v českém jazyce, s diakritikou si tedy Web CEO příliš hlavu neláme, čímž chybí hlavně možnost volby, zda diakritiku rozlišovat či nikoliv. Zároveň Web CEO neumožňuje rozšířené možnosti práce s textem, jako například synonyma, kořeny slov či seřazení slov v klíčovém slově podle abecedy. Na druhou stranu je díky tomu Web CEO rychlejší a díky tomu umožňuje stažení aplikace a používání jako desktopové, což při požadovaném výkonu mé aplikace při tvorbě všech kombinací klíčových slov není umožněno.

Dalším velkým nedostatkem Web CEO v porovnání s mou aplikací je porovnání výsledků s konkurenčními web. Web CEO porovnání sice obsahuje, ale ne porovnání on-page faktorů, nýbrž jen

porovnání toho, jak se na jednotlivé weby dívají vyhledávače. Opět protiváhou této funkčnosti je výkon, jelikož porovnání výsledků je výpočetně i paměťově velice náročné.

Nespornou výhodou pro Web CEO je určitě analýza off-page faktorů a vyhledávačů. Avšak, jak bylo již zmíněno, pracoval jsem v týmu, v němž ostatní kolegové měli za úkol implementovat právě tyhle funkce aplikace.

4.2 SpyFu

Bohužel, tato online aplikace přešla do plně zpoplatněného režimu. Avšak v minulosti jsem aplikaci testoval, i když jen v demo režimu, který nenabízel příliš mnoho možností. Nicméně SpyFu nabízí stejně jako moje aplikace generování klíčových slov nacházejících se na testované stránce. Stejně jako Web CEO nenabízí možnosti volby slovníků či diakritiky a pořadí slov. Aplikace je směrována pro amerického koncového zákazníka, takže co se týče diakritiky, není se čemu divit. Je možné, že v placené verzi podporuje SpyFu také vícero slovníků.

Další funkčností, ke které jsem se však v demo verzi nedostal, ale SpyFu tuto funkci propaguje je porovnání webové stránky s konkurencí. Do jaké míry však toto porovnání zasahuje, bohužel netuším. Nevím tedy, jestli se jedná o stejné porovnání jako v případě Web CEO, kdy jsou porovnávány jen údaje získané od vyhledávačů, nebo zda jde o podobné porovnání, které provádí moje aplikace, tedy porovnání on-page faktorů.

4.3 Seo Servis

Jedná se online aplikace v zastoupení českého prostředí. Aplikace tedy plně podporuje češtinu a navíc je zcela zdarma. Obsahuje hned několik typů testů stránky, úmyslně jsem zvolil „testů stránky“ jelikož aplikace postrádá jakékoliv porovnávání s konkurencí (pokud si ji neotevřeme ve dvou oknech) a „stránky“ pro to, že je aplikace zaměřená pouze na jednu stránku, nikoliv na celou doménu.

Obrázek 19: Náhled aplikace Seo Servis

4.3.1 Zdrojový kód

V téhle části aplikace kontroluje, zda jsou na stránce uvedeny všechny důležité tagy, validaci, apod. Jedná se o podobnou analýzu, ne-li stejnou, kterou moje aplikace nabízí v detailních informacích o stránce.

- Kontrola `doctype`, znakové sady, jazyku stránky a hlavičkových elementů `keywords`, `description`, `meta robots`, `meta author` a `title`
- Kontrola existence robots.txt a sitemap.xml
- Validace HTML kódu
- Kontrola velikosti HTML kódu
- Kontrola `alt` a `title` atributů u obrázků a odkazů
- Kontrola nadpisů `h1-h6`
- Kontrola počtu slov na stránce

– Nekontroluje jedinečnost nadpisů a hlavičkových tagů
– Nekontroluje dobu stažení stránky
– Nekontroluje počet překlepů na stránce
– Nevypisuje obsah jednotlivých elementů se zvýrazněním překlepů
– Nekontroluje délku nadpisů, URL adresy a hlavičkových tagů
\+ Kontrola, zda stránka obsahuje tabulky
\+ Kontrola struktury nadpisů

4.3.2 Klíčová slova

Zde nalezneme analýzu klíčových slov na námi zadané stránce. Analýza aplikace Seo Servis je opět poněkud strohá v porovnání s mou aplikací.

- Seznam klíčových slov na stránce
- Reálný výskyt každého klíčového slova na stránce a procentuální zastoupení
- Skóre neboli rank každého klíčového slova

– Chybí možnost zvolení slovníku, rozlišování diakritiky a pořadí slov
– Není možnost vybrat klíčové slova určité délky
– Nevidím všechny slova, jen prvních deset
– Není možnost nastavení přepočtu skóre u klíčových slov
– Nevpisuje text z HTML tagů se zvýrazněným klíčovým slovem
– Nenabízí rady, co udělat pro to, abychom zlepšili optimalizaci stránky na námi zadané klíčové slovo neboli kde přidat či ubrat klíčové slovo
\+ Vzhledem k tomu, že se analyzuje jen jedna stránka s jedním slovníkem, známe výsledek téměř okamžitě

4.3.3 Síla webu

Jedná se v podstatě o jakési zhodnocení výsledků zahrnujících také analýzu zdrojového kódu a vygenerování ranku stránky. Navíc jsou zde přidány údaje jako je PageRank či S-Rank, což nabízí i moje aplikace.

Co však moje aplikace neuvádí je pozice ve vyhledávačích, zpětné odkazy, počet indexovaných stránek vyhledávači a stáří domény. Je však třeba podotknout, že tohle již nespadá do on-page faktorů stránky, jedná se tedy o primární zaměření mojí aplikace. Avšak jak jsem již bylo zmíněno, kolegové z týmu na těchto funkcionalitách pracovali.

4.3.4 Závěrečné zhodnocení

Jedná se o aplikaci, která je zdarma a podporuje češtinu. To je pro českého uživatele určitě velké plus. Avšak v porovnání s mou aplikací je Seo Servis poněkud „hubenější“ co se nabízených funkcionalit týče. Jak jsem již zmínil, analyzuje pouze jednu stránku, takže pokud si někdo bude chtít zanalyzovat celý větší web, měl by si vyhradit dostatek času. Dále nenabízí porovnání výsledků s konkurencí, což v konečném důsledku může napovědět více, než analýza vlastních stránek.

5. Závěr

Hlavním cílem této práce bylo vytvořit aplikaci, která stáhne obsah celé domény a otestuje kvalitu těchto stránek z hlediska SEO a copywritingu. Získané výsledky potom měla aplikace porovnat s konkurenčními projekty nebo s ostatními stránkami stejné domény. Díky těmto analýzám může uživatel zhodnotit svůj vlastní projekt, opravit případné chyby či optimalizovat dle získaných rad aby se dostal ve výsledcích vyhledávání na internetu výše, než konkurence, se kterou svůj projekt porovnává.

Aby bylo možné tuto aplikaci implementovat, musel jsem nejprve detailně nastudovat problematiku SEO, především co se on-page faktorů stránky týče. Dále bylo vhodné nastudovat, jak fungují samotné vyhledávače, jelikož mnou vytvářená aplikace se snaží naznačit, jak vyhledávače vidí jednotlivé stránky. Při implementaci bylo tedy přínosné vědět, jak některé problémy řeší přední světové vyhledávače, u kterých jsem se poté mohl inspirovat.

Podmínkou aplikace bylo, aby stahování stránek probíhalo paralelně. Toho bylo docíleno díky implementace aplikace v programovacím jazyce Java. Navíc, když už bylo naimplementováno paralelní stahování, nic nebránilo tomu, aby aplikace i nadále pracovala paralelně. Jedná se tedy o aplikaci, která nejen paralelně stahuje stránky, ale také je paralelně parsuje, ukládá a následně porovnává s ostatními stránkami.

Požadavkem bylo také, aby aplikace nejen stahovala a parsovala stránky, ale aby také validovala její obsah. Pro řešení tohoto problému jsem využil existující Java knihovny, která validuje HTML kód dle W3C standardů. Navíc jsem aplikaci rozšířil o kontrolu překlepů na stránce, s využitím pravidel českého pravopisu.

Aplikace se také má soustředit na problematiku klíčových slov a sní spjatou problematiku copywritingu. Bylo tedy nutné, ze získaného obsahu stránky vygenerovat klíčové slova, které jsou následně analyzovány, a uživateli je předloženo řešení, jak docílit toho, aby byla stránka lépe optimalizována na jím požadované slovo. Navíc, co se týče práce s klíčovými slovy, aplikace obsahuje množství nastavení, které klíčové slova nás zajímají. Obsahuje tedy dohromady dvanáct různých slovníků, které kombinují možnosti klíčových slov jako, zda se mají skládat z originálních slov, synonym či kořenů slov, zda tyto jednotlivé slova v klíčovém slově řadit dle abecedy a zda rozlišovat diakritiku. Uživateli je tak nabídnuto hned několik pohledů na obsah jeho webových stránek a je jen na něm, který jej opravdu zajímá.

Jak jsem již zmiňoval, webové projekty a stránky jsou mezi sebou porovnávány. Toto porovnávání probíhá paralelně a jde především o porovnání SEO kvality stránky a klíčových slov na stránce, tedy jakési podobnosti webových projektů a jejich stránek. Při porovnávání podobnosti stránek se uživateli opět nabízí možnost volby slovníku a také modifikace výpočtu. Může si tedy například nastavit, zda jej zajímá porovnání slov které má navíc vůči konkurenční stránce či které mají společné. Je zde velké množství kombinací, avšak ne všechny musejí dávat relevantní výsledky. To už je však na posouzení uživatele, který si výpočet nastaví.

Jelikož je aplikace poměrně náročná na strojový čas, běží na serveru. Uživatel si zadá projekty, které chce analyzovat a po nějaké době si může projít výsledky. Jelikož jsou již výsledky přepočítány a uloženy v databázi, je jejich zobrazení v podstatě okamžité. K tomu, aby uživatel spustil projekt, potřebuje pouze webový prohlížeč, který podporuje jen základní funkčnosti běžného prohlížeče. Výsledky jsou uživateli přehledně nabídnuty v tabulkách se všemi informacemi, jenž je možné o stránce získat. Aby bylo zobrazení ještě přehlednější, aplikace si přepočítává SEO skóre pro jednotlivé stránky, čímž usnadní uživateli rozlišení kvalitních a nekvalitních stránek. Porovnání stránek je pak udáváno v % jako míra podobnosti, čímž uživatel ihned vidí, které stránky jsou si podobné.

Aplikace byla otestována asi na sto padesáti webových projektech a již úspěšně běží na školním serveru, kde čeká na webové projekty na zpracování. Jediným momentálním omezením aplikace je kapacita databáze, jelikož množství získaných informací z webových projektů je poměrně velké.

Bylo by vhodné vysledovat, které možnosti dávají relevantní výsledky, respektive které možnosti nejvíce zajímají uživatele aplikace a ostatní, nepoužívané možnosti nastavení eliminovat a šetřit tím tak jak místo v databázi, tak výpočetní výkon potřebný ke zpracování všech možností.

Do budoucna by bylo vhodným rozšířením aplikace, kdyby se navázala spolupráce s vyhledávačem Google, který by tak neblokoval větší množství dotazů na PageRank či zda je stránka indexována. Dále, by mohla být aplikace rozšířena pro desktopové užití, avšak bylo by nutné omezit množství slovníků, aby bylo zpracování rychlejší a méně paměťově náročné. Dalším možným rozšířením je procházení a validování CSS souborů a díky tomu kontrolovat velikost písma na stránce a barvu pozadí s barvou textu. Mohlo by se tak zabránit Black Hat SEO praktikám, o kterých uživatel občas ani neví, že jsou nekalé.

6. Seznam použité literatury

[1] KUBÍČEK, Michal. *Velký průvodce SEO*: *jak dosáhnout nejlepších pozic ve vyhledávačích*. Vyd. 1. Brno: Computer Press, 2008, 318 s. ISBN 978-80-251-2195-5.

[2] KUBÍČEK, Michal a Jan LINHART. *333 tipů a triků pro SEO*: *[sbírka nejlepších technik optimalizace webů pro vyhledávače]*. Vyd. 1. Brno: Computer Press, 2010, 262 s. ISBN 978-80-251-2468-0.

[3] GRAPPONE, Jennifer a Gradiva COUZIN. *SEO*: *search engine optimization: ovládněte SEO a získejte výhodu před konkurencí : optimalizujte své webové stránky pro vyhledávací servery : přiveďte na své stránky zákazníky dříve, než to udělá konkurence*. Vyd. 1. Brno: Computer Press, 2010, 262 s. ISBN 978-80-86815-85-5.

[4] HGENHAVEN, Edited by Rand Fishkin and Thomas. *Seomoz Guide to Inbound Marketing*. John Wiley, 2013. ISBN 11-185-5155-9.

[5] DOVER, Danny a Erik DAFFORN. *Search engine optimization secrets: do what you never thought possible with SEO*. Indianapolis, IN: Wiley, 2011, xix, 435 p. ISBN 978-111-8078-303.

[6] *Part II: Understanding Results - Google Guide* [online]. 2007 [cit. 2014-04-17]. Dostupné z: http://www.googleguide.com/category/understanding-results/

[7] *CRAWLING THE WEB: DISCOVERY AND MAINTENANCE OFLARGE-SCALE WEB DATA*. Stanford, 2001. Disertační práce. Stanford University.

[8] *Web metrics: Size and number of resources - Make the Web Faster – Google Developers* [online]. 2010 [cit. 2014-04-17]. Dostupné z: https://developers.google.com/speed/articles/web-metrics

[9] *The Anatomy of a Search Engine* [online]. 2006 [cit. 2014-04-17]. Dostupné z: http://infolab.stanford.edu/~backrub/google.html

[10] *Anatomy of a search engine : Infrastructure of Google* [online]. 2010 [cit. 2014-04-17]. Dostupné z: http://www.scriptol.com/web/google-anatomy.php

[11] *How a Search Engine Works* [online]. 2001 [cit. 2014-04-17]. Dostupné z: http://www.infotoday.com/searcher/may01/liddy.htm

[12] *Jak funguje Google našeptávač – Google Suggest* [online]. 2012 [cit. 2014-04-17]. Dostupné z: http://404m.com/2012/04/07/jak-funguje-google-naseptavac-google-suggest/

[13] *GOOGLE vs. SEZNAM.CZ | Effectix Doba Webová* [online]. 2013 [cit. 2014-04-17]. Dostupné z: http://www.doba-webova.com/cs/

[14] *Google PageRank, vysvětlení a odpovědi* [online]. 2014 [cit. 2014-04-17]. Dostupné z: http://www.jakpsatweb.cz/seo/pagerank.html

[15] *Robots.txt - zakázání přístupu robotům* [online]. 2014 [cit. 2014-04-17]. Dostupné z: http://www.jakpsatweb.cz/robots-txt.html

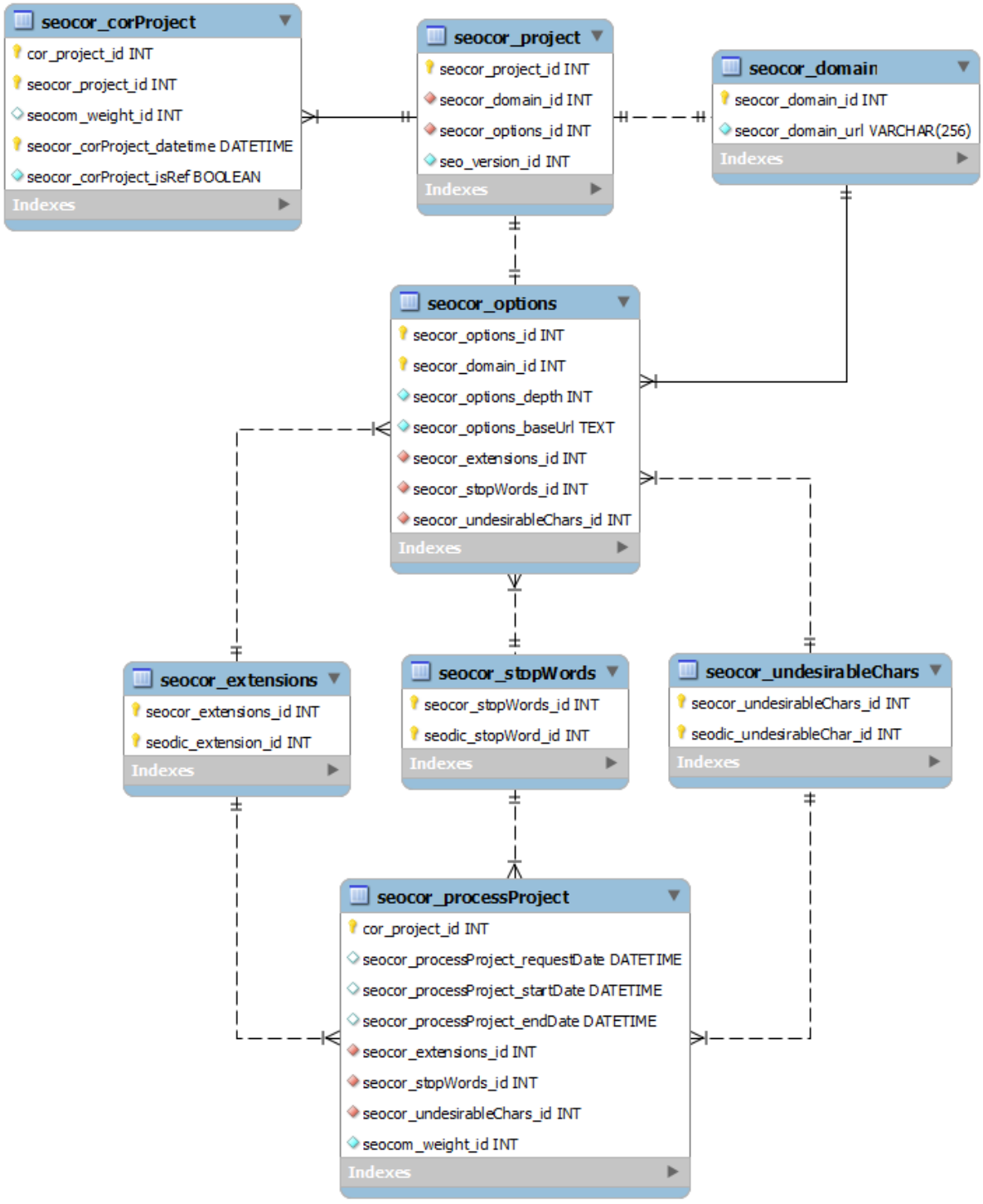
[16] *OneStat Website Statistics and website metrics - Press Room* [online]. 2006 [cit. 2014-04-17]. Dostupné z: http://www.onestat.com/html/aboutus_pressbox45-search-phrases.html

[17] *MySQL :: MySQL Workbench* [online]. 2014 [cit. 2014-04-17]. Dostupné z: http://www.mysql.com/products/workbench/

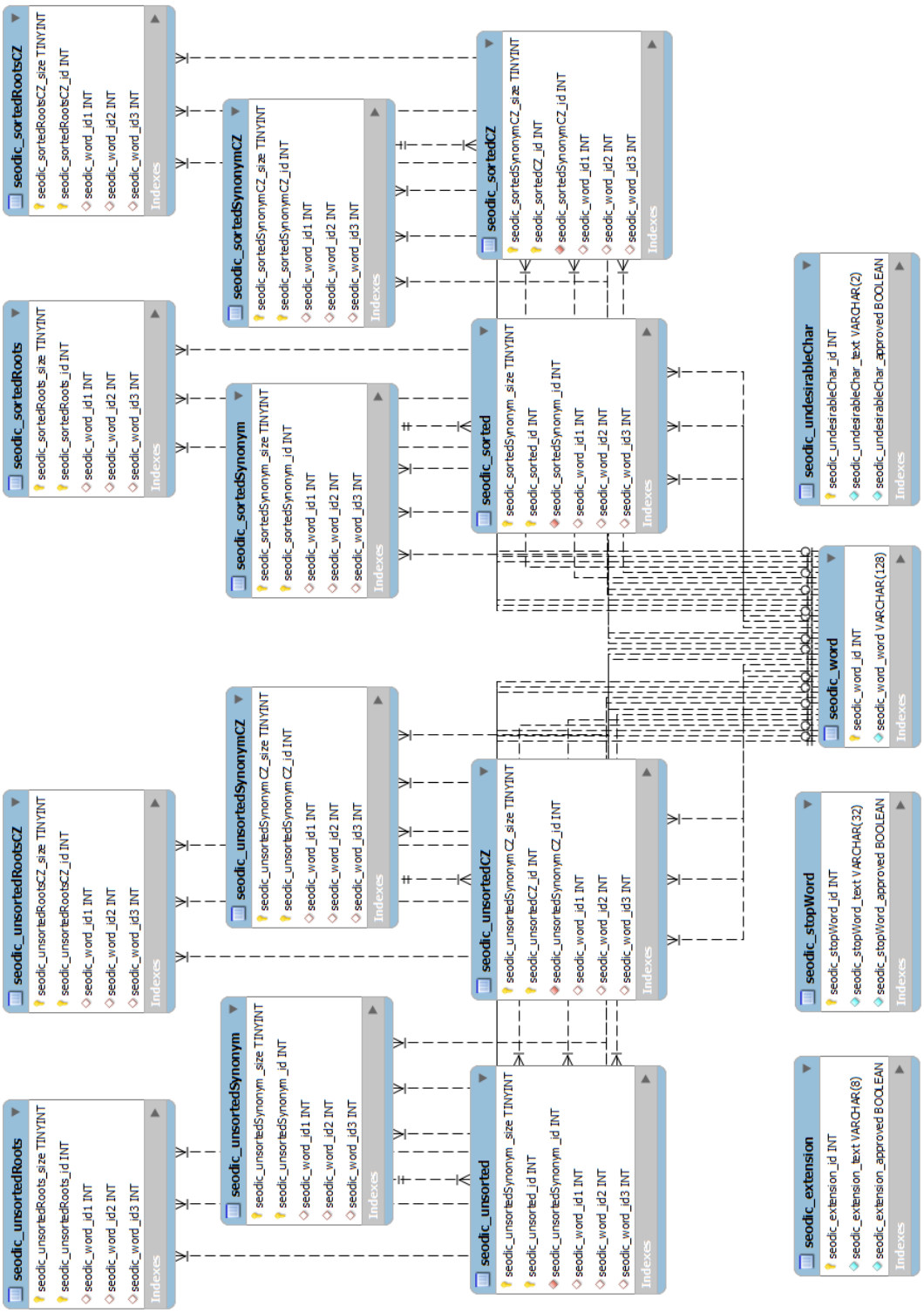
[18] *Microsoft Visio 2013 – flowchart software* [online]. 2014 [cit. 2014-04-17]. Dostupné z: http://office.microsoft.com/en-us/visio/

7. Seznam příloh

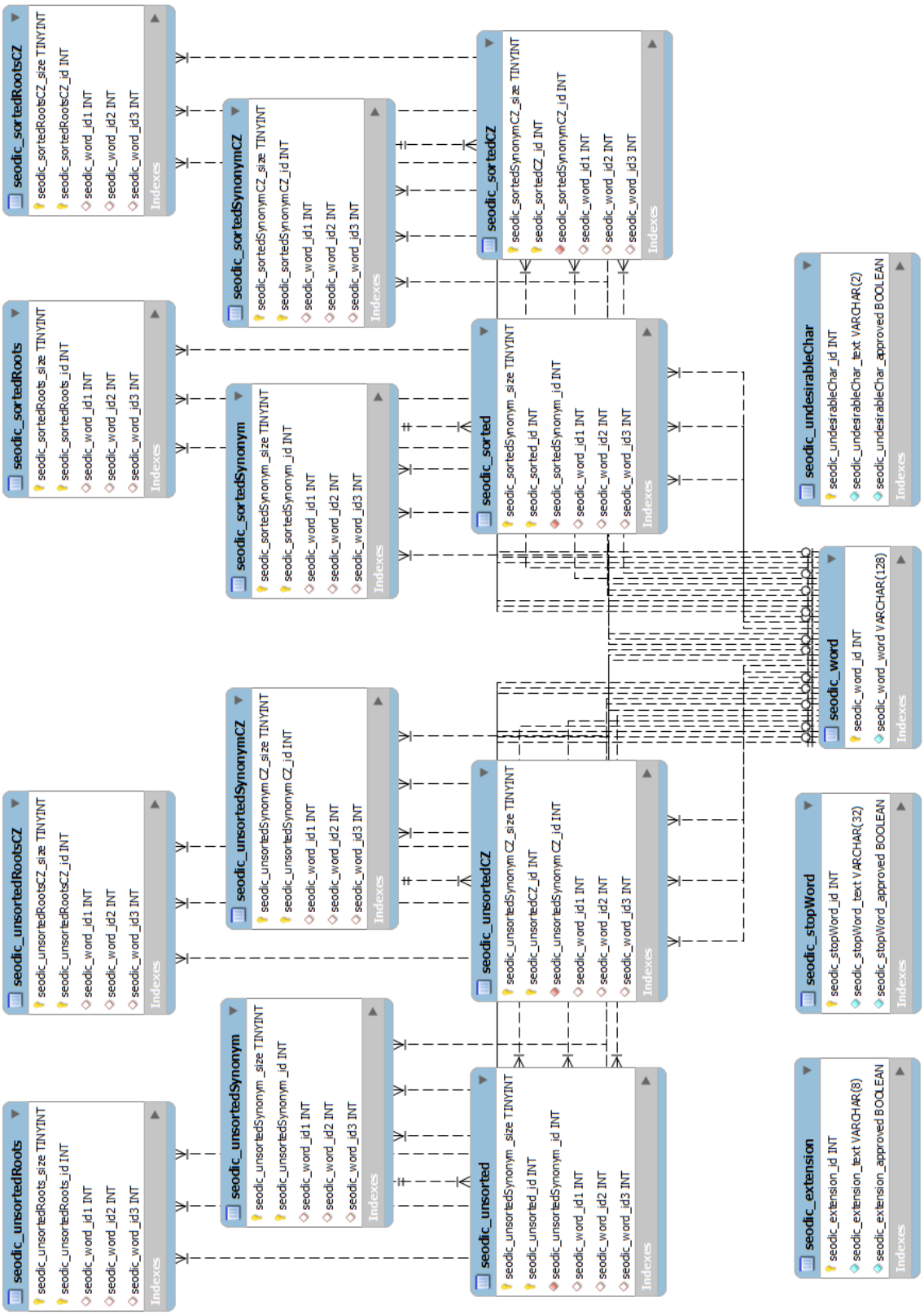
7.1 E-R Model - databáze SEOCOR



7.2 E-R Model - databáze SEODIC



7.2.1 E-R Model - databáze SEOCOM



7.3 E-R Model - databáze SEO